5-2007

# An Investigation of Nonlinear Estimation and System Design for Mechatronic Systems

Tim Hughes
*Clemson University*, tdhughe@clemson.edu

# AN INVESTIGATION OF NONLINEAR ESTIMATION AND SYSTEM DESIGN FOR MECHATRONIC SYSTEMS

---

A Thesis
Presented to
the Graduate School of
Clemson University

---

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Timothy Hughes
May 2007

---

Accepted by:
Dr. Darren Dawson, Committee Chair
Dr. John Wagner
Dr. Ian Walker

ABSTRACT

This thesis is a collection of two projects in which the author was involved during his master's degree program. The research involves the estimation of 3D Euclidean coordinates of features from 2D images. The research also includes a project funded by the Department of Energy (DOE) on the design of a test system for the Argon Environment Electrical Study (AEES).

The first chapter introduces both projects and provides the necessary background to the research. A review of the literature is presented for the estimation of 3D Euclidean coordinates from 2D images. Additionally, we provide a background to the issues associated with operating electrical equipment in an argon environment.

In the second chapter, we develop a 3D Euclidean position estimation strategy for a static object using a single moving camera whose motion is known. The Euclidean depth estimator which is developed has a very simple mathematical structure and is easy to implement. Numerical simulations and experimental results using a mobile robot in an indoor environment are presented to illustrate the performance of the algorithm.

In the third chapter, we describe the design of a test system for the AEES conducted by the DOE. We provide a summary of the completed design and describe how the final design was developed. The initial research proposal, safety

review, and literature review are presented. Additionally, the test plan and system design are highlighted.

In concluding the thesis, we discuss the performance of the 3D Euclidean position estimator versus the previous work and present possibilities for future work. We also discuss what has been learned from the design of the test system for the AEES. Finally, future studies for the AEES are offered.

# DEDICATION

This thesis is dedicated to my parents who by their love and encouragement have been an example of Christ (Romans 8:38-39).

# ACKNOWLEGEMENTS

This thesis would not be possible without the help and support from my advisors, colleagues, friends, and family. First, I would like to thank my advisor Dr. Darren Dawson for his motivation and guidance in the projects for which I was involved. I am also extremely grateful to Dr. John Wagner for reviewing my progress and giving valuable input each week for the argon study. I would like to thank all my committee members—Dr. Darren Dawson, Dr. John Wagner, and Dr. Ian Walker—for taking time to review my work.

I would like to thank especially David Braganza for his help and direction with the experiments. I am thankful for his meeting with me several times a week to discuss the experiments and giving suggestions. Additionally, the experiments would not be possible without the loan of a mobile robot from Dr. Stan Birchfield to whom I am very grateful.

I extend my gratitude to Don Erich and Eileen Brown for their involvement with the argon study. I am also grateful to Gary Youell for providing the opportunity to work on the argon study and to Charles Ralph for his helpfulness.

Finally, I would like to thank my roommates for the many memories we share from our time together at Clemson. I am also thankful for the many friends who prayed for me regularly. Above all, I am extremely grateful for the love and prayers from my wonderful family.

TABLE OF CONTENTS

Table of Contents (Continued)

Table of Contents (Continued)

Page

LIST OF TABLES

LIST OF FIGURES

List of Figures (Continued)

List of Figures (Continued)

# CHAPTER 1

# INTRODUCTION

## 1.1 Thesis Organization

This thesis consists of two sections each of which addresses a different problem and utilizes a separate approach. The first section is the development of a nonlinear estimator to calculate the Euclidean position of stationary features using a moving camera with known velocities. Much of the text from the first section comes from a work coauthored with David Braganza and Dr. Darren Dawson and submitted to the 46$^{th}$ IEEE Conference on Decision and Control. The second section features the design of a test system for the Argon Environment Electrical Study (AEES), a project funded by the Department of Energy (DOE).

## 1.2 Euclidean Position Estimation of Static Features using a Moving Camera with Known Velocities

The use of a camera to estimate the 3D structure of an object from 2D images is known as "Structure from Motion (SFM)" [1], [2], [3], or "Simultaneous Localization and Mapping (SLAM)" (see [4], [5], and references therein). The problem usually involves a camera mounted on a moving platform, such as an unmanned aerial vehicle (UAV) or a mobile robot, which is utilized to map the Euclidean position of static landmarks or visual features in the environment. Recently, SLAM and SFM have been utilized for a number of applications including aerial tracking and surveillance of ground based, stationary or moving objects [6], [7], [8], [9], and terrain mapping systems [10], [11], [12].

Most of the previous results in this area are formulated using linearization based techniques such as the extended Kalman filter [3], [4], [5]. It has been noted [13] that the linearized motion models can cause significant inconsistencies in solutions. There have been a few results [14], [15], [16], which utilized nonlinear system analysis and estimation tools to design nonlinear observers for the problem. In recent work, Chitrakaran et al. [17], [18] proposed nonlinear estimation strategies to identify the Euclidean structure of an object using a monocular calibrated moving camera. The camera motion in this work was modeled based on the homography between two different views captured from the camera, the current frame and a constant reference frame. The algorithms reported by Chitrakaran et al., require that at least one distance between two features on the object be known for the reconstruction of the 3D Euclidean coordinates. Also, to decompose the homography and obtain the rotation and translation of the camera between the two camera views, the normal vector to the object must be known [18] and in the case of [17], the rotation between the object frame and the camera at the reference position must also be known.

In this work, our objective is to estimate the 3D Euclidean structure of a static object using a single camera mounted on a moving platform whose translation and rotation velocities are measurable. Although the work in [17], [18], was fundamentally more challenging, since the camera velocity was unknown, it did make some assumptions on the structure of the object which it was to identify. There are applications such as video surveillance and mapping using a UAV or a mobile robot where the velocity of the camera mounted on the moving platform is

readily available. Thus, the goal of this work is to eliminate the requirements from the previous works [17], [18], that the distance between two feature points be known, and that the normal vector or rotation matrix be known a priori. The development in this work is similar to the concepts introduced in [19], [20], where range observer's were developed for feature points on an object undergoing affine motion with known motion parameters. However, the development in our work is based on the kinematics of the moving camera and has a simpler mathematical formulation.

To design the estimator, the equations for motion kinematics are first developed in terms of Euclidean and imagespace information based on a single camera view [21]. Then, a nonlinear integral observer [22], is utilized to estimate the velocity of each feature point in the image plane. Once the estimate of the image velocity is known a simple estimator can be developed for the depth variable, and hence, the 3D structure can be estimated. The developed estimator asymptotically identifies the Euclidean depth subject to an observability condition. This condition is similar to the observability condition of [19], [20] and the persistency of excitation condition in [18]. The proposed estimator was implemented using a camera mounted on a mobile robot and our experimental results show that the estimator converges very quickly and is not computationally complex, and hence, can be used for real-time applications.

### 1.3 Design of Test System for the Argon Environment Electrical Study

The purpose of the AEES is to determine the effects of operating electric motors and electrical connectors in an argon environment. This project is with the

Clemson Engineering Technologies Laboratory (CETL). The project customer is the Department of Energy (DOE) at the Savannah River Site (SRS).

There are several issues to consider when operating electrical equipment in an argon atmosphere. The first is electrical arcing. Compared to air, argon has a lower dielectric constant. For this reason, the voltage at which electrical breakdown or arcing occurs is lower. There are two places where the electrical breakdown can occur. The first is in the electrical connectors. The other area where breakdown can occur is inside the motors.

Another issue to consider is that the heat transfer ability of argon is lower than that of air. When operating electric motors in a self-contained glovebox, the heat from the motors causes the temperature inside the glovebox to rise. The increase in temperature will damage the motors unless the motors are equipped with sensors that turn off the motors when the temperature exceeds a safe limit.

# CHAPTER 2

# EUCLIDEAN POSITION ESTIMATION OF STATIC FEATURES USING A MOVING CAMERA WITH KNOWN VELOCITIES

## 2.1 Overview

This chapter is organized as follows, in Section 2.2, the geometric model which relates Euclidean coordinates of visual features on the stationary object with their corresponding image pixel coordinates is developed based on the perspective projection model. Section 2.3, describes the motion kinematics between the camera and the object. Section 2.4, describes the velocity estimator which is used to estimate the pixel coordinate velocity of the visual features, and in Section 2.5, the Euclidean depth estimator is developed. Finally, numerical simulation and experimental results using a mobile robot in an indoor environment are presented in Section 2.6 and Section 2.7, respectively.

## 2.2 Geometric Model

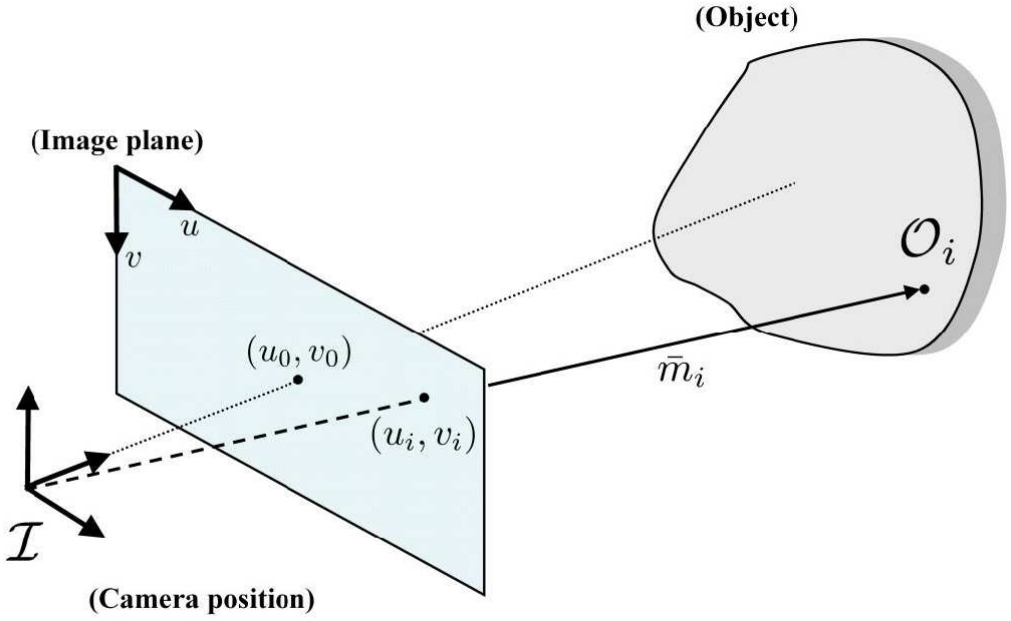


**Figure 2.1: Geometric model**

Figure 2.1 shows the geometric relationship between a moving perspective camera and features on a static object in its field of view. The geometric model developed in this section is based on a single view of the object from the camera at a time varying position denoted by $\mathcal{I}$. The vector $\bar{m}_i \in \mathbb{R}^3$ denotes the 3D Euclidean position of the $i^{th}$ feature point $O_i$ relative to the camera frame $\mathcal{I}$, and is defined as

$$\bar{m}_i \triangleq \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T. \tag{2.1}$$

the moving camera is the normalized Euclidean coordinates of the feature points, denoted by $m_i \in \mathbb{R}^3$ which is defined as

$$m_i \triangleq \frac{1}{z_i} \bar{m}_i. \tag{2.2}$$

The corresponding projective pixel coordinates of the feature points are denoted by $p_i \in \mathbb{R}^3$ which is defined as

$$p_i \triangleq \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T . \tag{2.3}$$

The image coordinates of the features and their normalized Euclidean coordinates are related by the pin-hole camera model [21] such that

$$p_i = Am_i \tag{2.4}$$

where $A \in \mathbb{R}^{3\times3}$ is a known, constant, and invertible intrinsic camera calibration matrix defined as [23]

$$A = \begin{bmatrix} fk_u & fk_u \cot(\phi) & u_0 \\ 0 & \dfrac{fk_v}{\sin(\phi)} & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.5}$$

where $u_0, v_0 \in \mathbb{R}$ denote the pixel coordinates of the principal point (i.e., the image center that is defined as the frame buffer coordinates of the intersection of the optical axis with the image plane), $k_u, k_v \in \mathbb{R}$ represent camera scaling factors, $\phi \in \mathbb{R}$ is the angle between the camera axes, and $f \in \mathbb{R}$ denotes the camera focal length.

### 2.3 Camera Kinematics

The kinematics of the camera frame $\mathcal{I}$ is developed in terms of the image coordinates of the feature points. After taking the time derivative of Eq. (2.4), the following kinematics can be obtained (see [24] for more details)

$$\dot{p}_i = -\frac{1}{z_i} A_{ei} v_c + A_{ei} \left[ A^{-1} p_i \right]_\times \omega_c \tag{2.6}$$

where $v_c, \omega_c \in \mathbb{R}^3$ denote the translational and rotational velocities of the camera relative to the initial position of the camera but expressed in the local frame $\mathcal{I}$, and $A_{ei} \in \mathbb{R}^{3\times3}$ is a function of the intrinsic camera calibration matrix and the image coordinates of the $i^{th}$ feature points image coordinates, defined as

$$A_{ei} = A - \begin{bmatrix} 0 & 0 & u_i \\ 0 & 0 & v_i \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.7}$$

and $[\varsigma]_\times$ denotes the following skew-symmetric matrix

$$[\varsigma]_\times = \begin{bmatrix} 0 & -\varsigma_3 & \varsigma_2 \\ \varsigma_3 & 0 & -\varsigma_1 \\ -\varsigma_2 & \varsigma_1 & 0 \end{bmatrix} \quad \forall \ \varsigma = \begin{bmatrix} \varsigma_1 \\ \varsigma_2 \\ \varsigma_3 \end{bmatrix}. \tag{2.8}$$

For the remainder of this development only the first two elements of $\dot{p}_i(t)$ defined in Eq. (2.6) are considered. Thus, the 2D kinematics for the camera can be written as

$$\dot{X}_i = -\frac{1}{z_i}\Pi_i v_c + \Pi_i \left[A^{-1} p_i\right]_\times \omega_c \tag{2.9}$$

where $X_i(t) \in \mathbb{R}^2$ is expressed as

$$X_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^T \tag{2.10}$$

and $\Pi_i \in \mathbb{R}^{2\times3}$ consists of the first two rows of the matrix $A_{ei}$ which was defined in Eq. (2.7), and can be explicitly written as

$$\Pi_i = \begin{bmatrix} fk_u & fk_u \cot(\phi) & u_0 - u_i \\ 0 & \dfrac{fk_v}{\sin(\phi)} & v_0 - v_i \end{bmatrix}. \tag{2.11}$$

## 2.4 Image Feature Velocity Estimation

The only unknown in the camera kinematic equation Eq. (2.9), is the Euclidean depth $z_i(t)$. To facilitate the development of an estimator for the depth parameter, an estimate of the image velocity signal $\dot{X}_i(t)$ is required. The following continuous estimator [22] can be utilized to estimate the velocity

$$\dot{\hat{X}}_i \triangleq \int_{t_0}^{t} \left[ (K_1 + I_2)\tilde{X}_i(\tau) + K_2 \operatorname{sgn}\left(\tilde{X}_i(\tau)\right) \right] d\tau + (K_3 + I_2)\tilde{X}_i(t) \qquad (2.12)$$

where $\dot{\hat{X}}_i \triangleq \left[ \dot{\hat{u}}_i \quad \dot{\hat{v}}_i \right]^T \in \mathbb{R}^2$ denotes the estimate of the signal $\dot{X}_i(t), \tilde{X}_i(t) \in \mathbb{R}^2$ is the estimation error defined as

$$\tilde{X}_i(t) \triangleq X(t) - \hat{X}(t), \qquad (2.13)$$

$K_1, K_2, K_3 \in \mathbb{R}^{2\times 2}$ denote constant positive definite diagonal gain matrices, and $\operatorname{sgn}\left(\tilde{X}_i\right)$ denotes the signum function applied to each element of the vector $\tilde{X}_i(t)$. For more details on the development of the above estimator the reader is referred to [22]. To summarize the result, it was shown that the estimator in Eq. (2.12) asymptotically identifies the signal $\dot{X}_i(t)$ (i.e. $\left\| \dot{\tilde{X}}_i(t) \right\|, \left\| \tilde{X}_i(t) \right\| \to 0$ as $t \to \infty$), provided that the $j^{th}$ diagonal element of the gain matrix $K_2$ and the $j^{th}$ element of the vectors $\ddot{X}_i(t)$ and $\dddot{X}_i(t)$ satisfies the following condition for all *i* feature points

$$[K_2]_j \geq \left| \left[ \ddot{X}_i(t) \right]_j \right| + \left| \left[ \dddot{X}_i(t) \right]_j \right| \quad \forall j = 1, 2 . \qquad (2.14)$$

Thus the only restriction on the camera motion is a relatively mild assumption of the smoothness and boundedness of the higher order derivatives of the camera velocity.

## 2.5 Euclidean Depth Estimation

The objective is to design an estimator for the Euclidean depth, $z_i(t)$. To this end, the kinematic equation Eq. (2.9), can be rewritten in a simplified form as

$$\dot{X}_i = -\rho_i \lambda_i + \delta_i \tag{2.15}$$

where $\lambda_i = \begin{bmatrix} \lambda_{i1} & \lambda_{i2} \end{bmatrix}^T \in \mathbb{R}^2$, $\delta_i = \begin{bmatrix} \delta_{i1} & \delta_{i2} \end{bmatrix}^T \in \mathbb{R}^2$ are measurable signals which are defined as

$$\lambda_i = \Pi_i v_c \tag{2.16}$$

$$\delta_i = \Pi_i \left[ A^{-1} p_i \right]_\times \omega_c \tag{2.17}$$

and $\rho_i(t) = \dfrac{1}{z_i(t)} \in \mathbb{R}$ is the inverse of the Euclidean depth which is unknown and will be estimated.

The individual components of the simplified expression for the camera kinematics in Eq. (2.15), can be written as

$$\dot{\tilde{X}}_{i1} + \dot{\hat{X}}_{i1} = -\rho_i \lambda_{i1} + \delta_{i1} \tag{2.18}$$

$$\dot{\tilde{X}}_{i2} + \dot{\hat{X}}_{i2} = -\rho_i \lambda_{i2} + \delta_{i2} \tag{2.19}$$

where $\tilde{X}_i = \begin{bmatrix} \tilde{X}_{i1} & \tilde{X}_{i2} \end{bmatrix}^T$, $\hat{X}_i = \begin{bmatrix} \hat{X}_{i1} & \hat{X}_{i2} \end{bmatrix}^T$, and Eq. (2.13) was utilized. After multiplying Eq. (2.18) by $\lambda_{i1}(t)$ and Eq. (2.19) by $\lambda_{i2}(t)$, and rearranging the resulting equations, the following expressions can be obtained

$$\rho_i \lambda_{i1}^2 = \lambda_{i1} \left( \delta_{i1} - \dot{\hat{X}}_{i1} \right) - \lambda_{i1} \dot{\tilde{X}}_{i1} \tag{2.20}$$

$$\rho_i \lambda_{i2}^2 = \lambda_{i2} \left( \delta_{i2} - \dot{\hat{X}}_{i2} \right) - \lambda_{i2} \dot{\tilde{X}}_{i2} \tag{2.21}$$

After adding Eq. (2.20) and Eq. (2.21), the following expression is obtained

$$\rho_i\left(\lambda_{i1}^2+\lambda_{i2}^2\right)=\lambda_{i1}\left(\delta_{i1}-\dot{\hat{X}}_{i1}\right)+\lambda_{i2}\left(\delta_{i2}-\dot{\hat{X}}_{i2}\right)-\lambda_{i1}\dot{\tilde{X}}_{i1}-\lambda_{i2}\dot{\tilde{X}}_{i2}. \tag{2.22}$$

Based on the expression in Eq. (2.22), an estimate for the inverse Euclidean depth can be designed as

$$\hat{\rho}_i \triangleq \frac{1}{\left(\lambda_{i1}^2+\lambda_{i2}^2\right)}\left[\lambda_{i1}\left(\delta_{i1}-\dot{\hat{X}}_{i1}\right)+\lambda_{i2}\left(\delta_{i2}-\dot{\hat{X}}_{i2}\right)\right]. \tag{2.23}$$

where $\hat{\rho}_i(t)\in\mathbb{R}$ represents the inverse depth estimate and the inverse depth estimation error $\tilde{\rho}_i(t)\triangleq\rho_i(t)-\hat{\rho}_i(t)\in\mathbb{R}$ is explicitly defined as

$$\tilde{\rho}_i=\frac{-1}{\left(\lambda_{i1}^2+\lambda_{i2}^2\right)}\left[\lambda_{i1}\dot{\tilde{X}}_{i1}+\lambda_{i2}\dot{\tilde{X}}_{i2}\right]. \tag{2.24}$$

Notice that, since the image feature velocity estimator asymptotically converges to the true velocity (i.e., $\dot{\tilde{X}}_{i1}(t),\dot{\tilde{X}}_{i2}(t)\to 0$), the inverse depth estimation error converges to zero, (i.e., $\tilde{\rho}_i(t)\to 0$). Thus, the inverse depth estimate $\hat{\rho}_i(t)$, converges to its true value provided that, $\dot{\hat{X}}(t)\to\dot{X}(t)$ and $\left(\lambda_{i1}^2(t)+\lambda_{i2}^2(t)\right)\neq 0$. From Eq. (2.15), it is evident that, if $\left(\lambda_{i1}^2(t)+\lambda_{i2}^2(t)\right)=0$, then the inverse depth estimate $\rho_i(t)$ is unobservable. Thus, we can conclude that the inverse depth estimate can be asymptotically identified provided that $\left(\lambda_{i1}^2(t)+\lambda_{i2}^2(t)\right)\neq 0$ and the gain condition in Eq. (2.14) is satisfied.

## 2.6 Simulation Results

A simulation study was conducted to evaluate the performance of the proposed estimation algorithm. The Simulink model used for the simulation is described in

Appendix A. The simulations were performed using five static feature points whose Euclidean coordinates were selected as follows:

$$\begin{aligned} O_1 &= \begin{bmatrix} 0 & 0.2 & 1 \end{bmatrix}^T \\ O_2 &= \begin{bmatrix} -0.1 & 0.2 & 1.25 \end{bmatrix}^T \\ O_3 &= \begin{bmatrix} 0.1 & 0.2 & 1.5 \end{bmatrix}^T \\ O_4 &= \begin{bmatrix} -0.2 & 0.2 & 1.75 \end{bmatrix}^T \\ O_5 &= \begin{bmatrix} 0.2 & 0.2 & 2 \end{bmatrix}^T \end{aligned} \tag{2.25}$$

The camera's translational and rotational velocities were chosen as

$$\begin{aligned} v_c(t) &= \begin{bmatrix} 0.2\cos(t) & 0.2\sin(t) & 0.1\sin(t) \end{bmatrix}^T \\ \omega_c(t) &= \begin{bmatrix} 0 & 0 & 0.1\sin(0.2\pi \cdot t) \end{bmatrix}^T \end{aligned} \tag{2.26}$$

In addition, a camera calibration matrix for a $640 \times 480$ camera was selected as follows:

$$A = \begin{bmatrix} 810 & 0 & 320 \\ 0 & 820 & 240 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.27}$$

The estimator gains were chosen to give the best performance both with and without additive noise and were selected as follows:

$$K_1 = diag\{3,3\}, K_2 = diag\{3,3\}, K_3 = diag\{20,20\} \tag{2.28}$$

In the simulations, four different cases were considered using the above parameters. For case 1, the image points had no noise added to them. In case 2, a small amount of noise (variance 0.001) was added to the image points. For case 3, noise with a variance of 0.001 was added and image points were passed through a low-pass filter. The low-pass filter had a cutoff frequency of 2 Hz. In the final case, the image points were rounded to integers to simulate the discrete output of

the feature tracker, and these image points were then passed through the low-pass filter.

The simulation results for each of the four cases are shown in Table 2.1. Note that feature points that are farther from the camera generally have a larger error. The highest percent error was 3.8% for case 2 with the feature point at a distance of 2 m from the camera. The depth estimation error for the four cases considered in the simulations is shown in Figures 2.2, 2.3, 2.4, 2.5. Figure 2.6 shows a comparison of the depth estimation error using the current algorithm and the algorithm from [18] for a single feature point.

**Table 2.1: Maximum error in depth estimation for each case in simulation**

| Feature Point | Depth (cm) | Error Case 1 (cm) | Error Case 2 (cm) | Error Case 3 (cm) | Error Case 4 (cm) |
|---|---|---|---|---|---|
| $O_1$ | 100 | 1.49 | 2.76 | 1.48 | 1.60 |
| $O_2$ | 125 | 1.85 | 3.71 | 2.00 | 2.26 |
| $O_3$ | 150 | 1.96 | 4.80 | 2.27 | 2.62 |
| $O_4$ | 175 | 2.43 | 6.58 | 2.65 | 3.36 |
| $O_5$ | 200 | 2.64 | 7.59 | 3.77 | 5.24 |

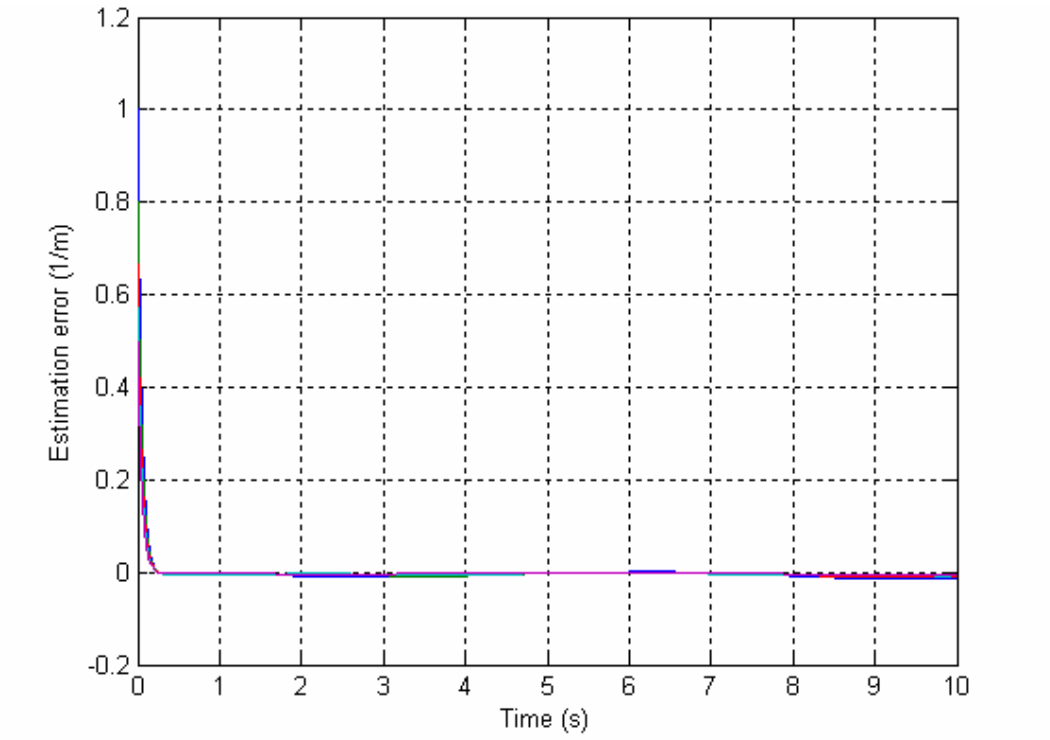


**Figure 2.2: Simulation case 1 – Depth estimation error without additive noise**

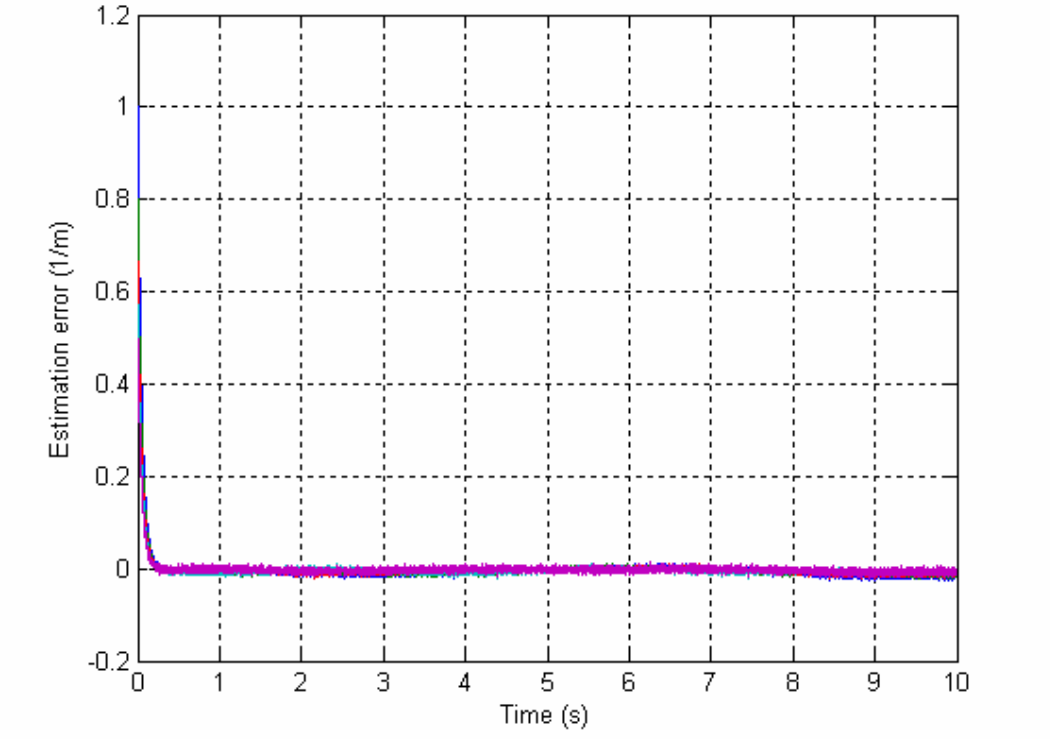


**Figure 2.3: Simulation case 2 – Depth estimation error with noise of variance 0.001**

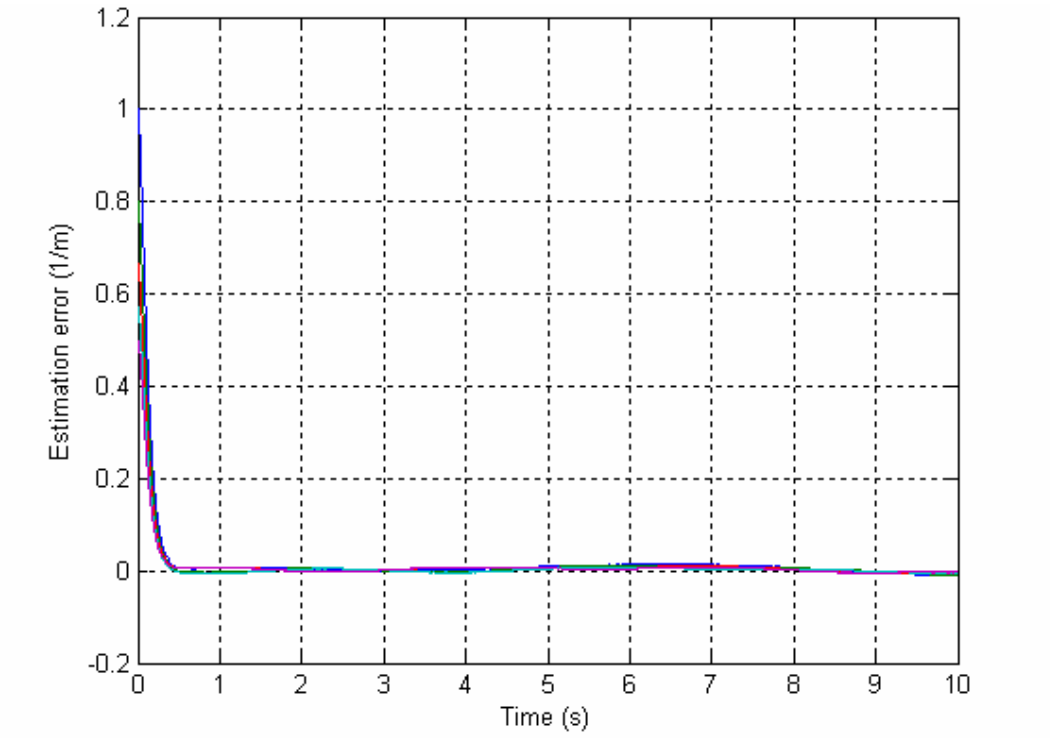


**Figure 2.4: Simulation case 3 – Depth estimation error with noise of variance 0.001 and filtering**

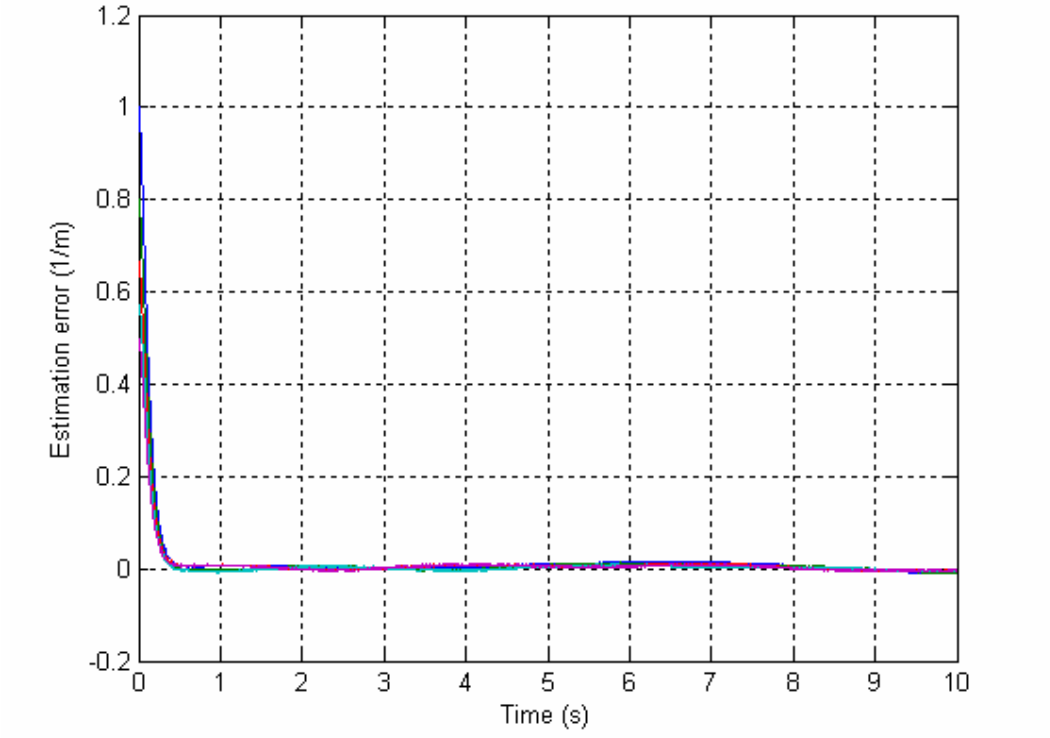


**Figure 2.5: Simulation case 4 – Depth estimation error with integer rounding**

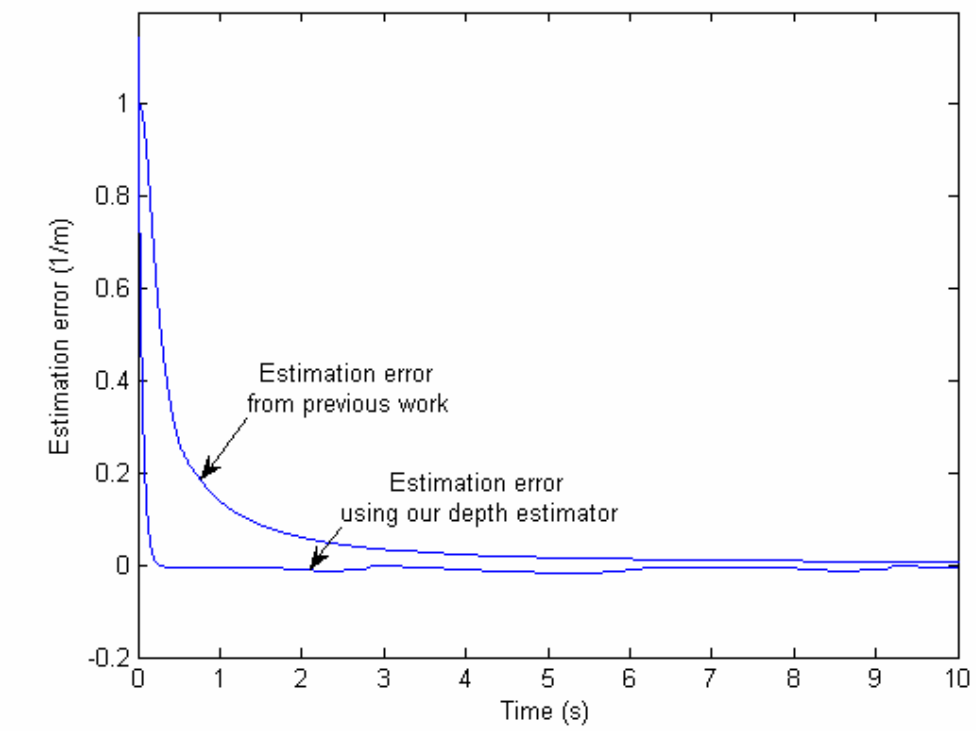


**Figure 2.6: Simulation comparison of the depth estimation error for a single feature point**

## 2.7 Experimental Results

### *2.7.1 Webcam Experimental Results*

In this section, experimental results using a mobile robot are discussed. A standard off the shelf webcam (Logitech QuickCam) was used to capture images at a resolution of 640 x 480 pixels. The calibration matrix of the camera was found to be the following

$$A = \begin{bmatrix} 762.6 & 0 & 333.3 \\ 0 & 760.8 & 226.2 \\ 0 & 0 & 1 \end{bmatrix}. \qquad (2.29)$$

The camera was mounted on an ActivMedia Robotics Pioneer 3 mobile robot as shown in Figure 2.7. The mobile robot's on-board controller provides translational and rotational velocity information using wheel mounted optical encoders at a rate of 10 samples per second. The test scene consisted of a dollhouse. Both the mobile robot and the camera were connected to a laptop with an Intel Centrino Duo 2 GHz processor and 1 GB of memory. The laptop was used to set the velocity of the robot, capture images of the scene, and log the video and velocity data for off-line processing. The robot was given a translational velocity of 5 cm/s along the *x*-axis and no rotational velocity. The average frame rate obtained using the webcam was 14.2 frames per second.

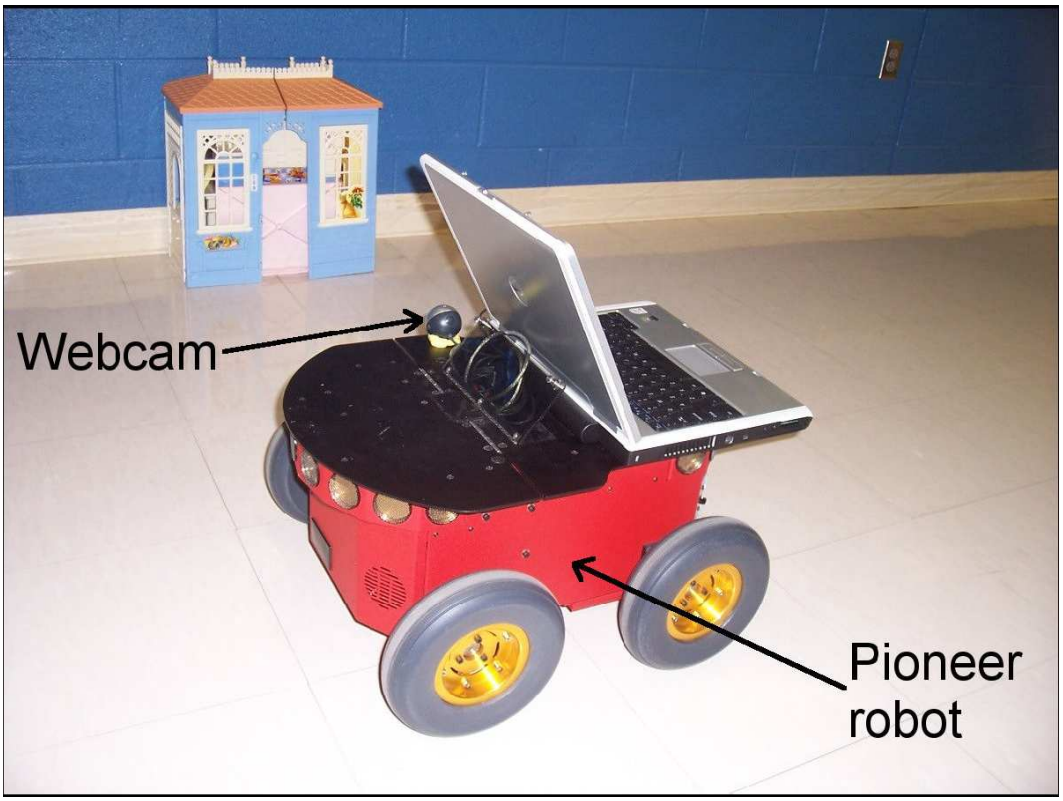


**Figure 2.7: Experimental test setup with camera, mobile robot, and dollhouse scene**

Using the implementation of the Lucas-Kanade feature tracking algorithm provided in the OpenCV computer vision library [25], a computer program was written in C++ which enabled the user to select features manually and track those features for the entire image sequence. The program created a text file which contained the feature point pixel coordinates and camera velocity for each frame. In the experiment, twelve features were selected. A sample frame with the tracked feature points is shown in Figure 2.8.

**Figure 2.8: A frame from the dollhouse image sequence showing the tracked feature points**

The depth estimation was calculated off-line using Mathworks Simulink program. A low-pass filter with a cutoff frequency of 1 Hz was applied to the feature points and camera velocities from the text file to smooth the data. The following velocity estimator gains were used:

$$K_1 = diag\{1,1\}, K_2 = diag\{1,1\}, K_3 = diag\{5,5\} \tag{2.30}$$

The estimated distance between features is shown in Figure 2.9. Note that the estimated values stabilize in under 1 second. The estimation error is shown in Figure 2.10. To illustrate how the image velocity estimator is useful, the image velocity estimator was replaced with a derivative operator. The distance estimation error was seen to be much higher without the image velocity estimator.

In fact, the estimation error more than doubled when the velocity estimator was replaced with the simple derivative operation. Figure 2.11 shows the estimated distance between features, and Figure 2.12, shows the estimation error for the case with the derivative operation.

**Table 2.2: Webcam experiment – Experimental error in distance estimation**

| Object | Actual Distance (cm) | Maximum Error (cm) | Percent Error |
|---|---|---|---|
| Length I | 10.0 | 1.44 | 14.4% |
| Length II | 23.7 | 2.86 | 12.1% |
| Length III | 40.0 | 4.56 | 11.4% |
| Length IV | 33.7 | 4.11 | 12.2% |
| Length V | 24.5 | 2.72 | 11.1% |
| Length VI | 24.5 | 2.88 | 11.8% |

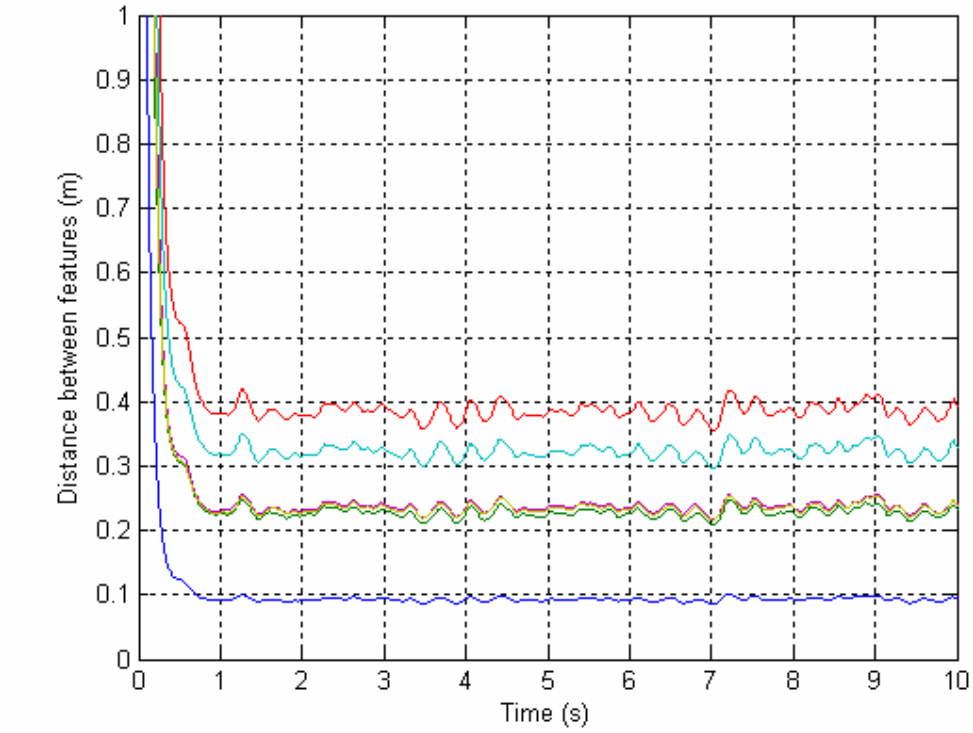


**Figure 2.9: Webcam experiment – Estimated distance between features**

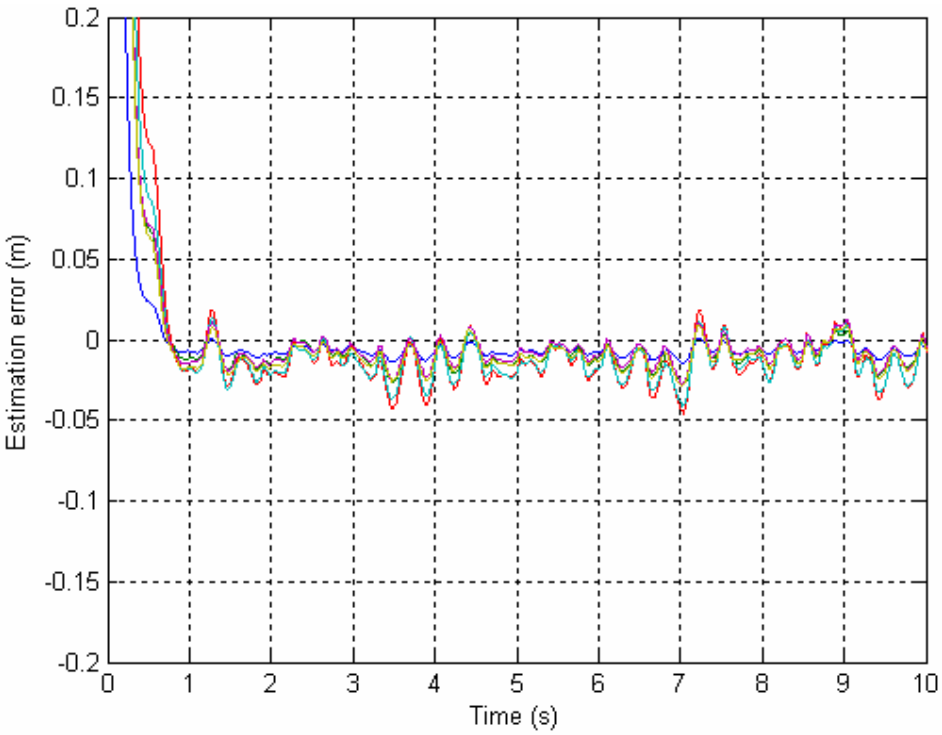


**Figure 2.10: Webcam experiment – Distance estimation error**

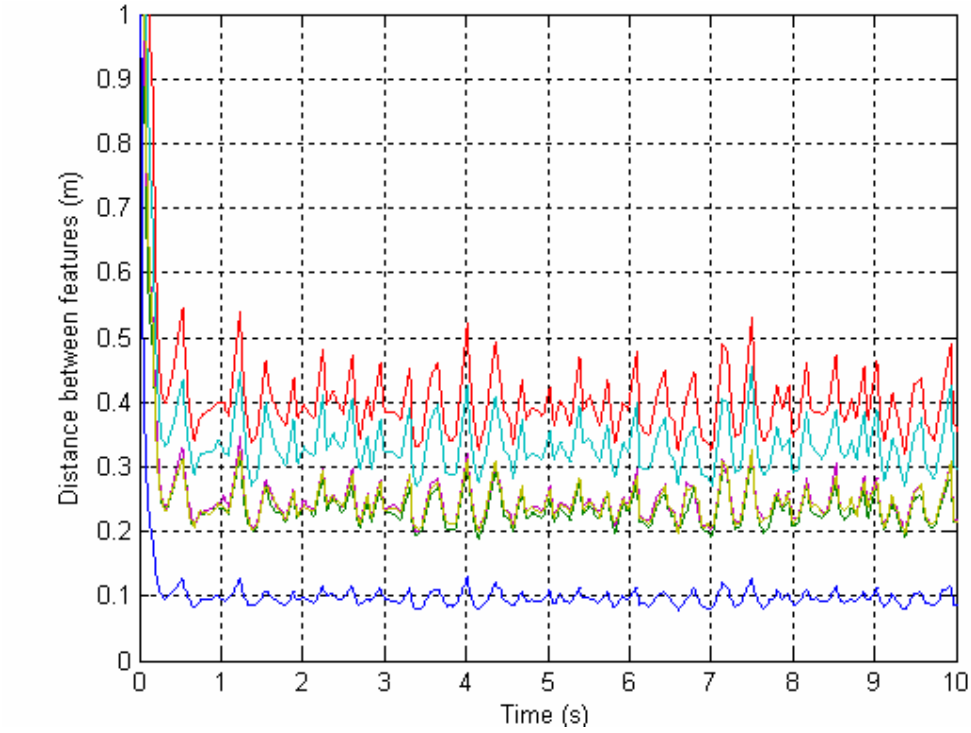


**Figure 2.11: Webcam experiment – Estimated distance between features with derivative operator**

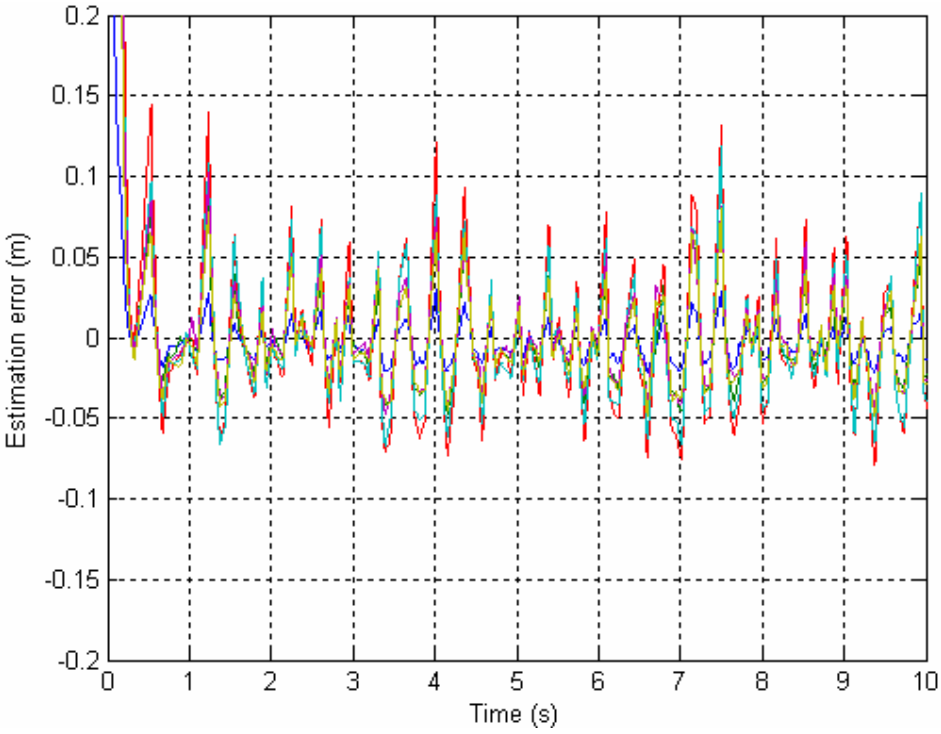


**Figure 2.12: Webcam experiment – Distance estimation error with derivate operator**

### *2.7.2 High-Definition Camera Experimental Results*

Overview

This section discusses experimental results using a high-definition (HD) camera. The purpose of this experiment was to investigate any improvements when using a camera with a higher resolution and faster frame rate. In place of the webcam, a Sony Handycam HDR-HC1 was mounted on the mobile robot and used to capture video of the scene. The video was captured at the standard NTSC frame rate of 29.97 frames per second. Using Adobe Premiere Pro, the video was deinterlaced and converted to individual frames having a resolution of 1440 x 810 pixels. The camera calibration matrix for this camera was found to be

$$A = \begin{bmatrix} 1585 & 0 & 772.4 \\ 0 & 1576.8 & 332.7 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.31}$$

The mobile robot was connected once again to a laptop. However, for this experiment the laptop was used only to set the velocity of the robot and to log the velocity data for off-line processing. The dollhouse was used again for the scene. As before, 12 features were tracked with the Lucas-Kanade feature tracking algorithm provided in OpenCV, and the depth estimation was calculated off-line using Simulink. Two different trajectories were tested—a straight line trajectory and a sinusoidal trajectory. In both cases, the direction of the camera was along the positive *y*-axis of the robot, whereas the motion was primarly along the positive *x*-axis of the robot (see Figure B.1).

In the webcam experiment, the magnitude of the robot velocity was equal to the magnitude of the camera velocity because the motion was in one direction (no rotational velocities) and the *x*-axis of the robot was parallel to the *x*-axis of the camera. In this HD camera experiment, the robot can have both translational and rotational velocities. Appendix B discusses the camera calibration and explains how a transformation from the robot reference frame to the camera reference frame was found. This transformation was necessary to convert the robot velocities to camera velocities.

Straight-line Trajectory

For the straight line trajectory, the robot was given a translational velocity along the *x*-axis of 5 cm/s and no rotational velocity. A low-pass filter with a cutoff frequency of 0.2 Hz was applied to the feature points and camera velocities from the text file to smooth the data. The following velocity estimator gains were used:

$$K_1 = diag\{1,1\}, K_2 = diag\{1,1\}, K_3 = diag\{5,5\} \tag{2.32}$$

Figure 2.13 shows the camera velocities obtained from the robot velocities measured during the experiment. The distance between feature points is shown in Figure 2.14 and the estimation error in the distance between feature points is shown in Figure 2.15. The maximum estimation error and percent error is listed in Table 2.3.

**Table 2.3: HD camera experiment – Experimental error in distance estimation for straight-line trajectory**

| Object | Actual Distance (cm) | Maximum Error (cm) | Percent Error |
|---|---|---|---|
| Length I | 9.8 | 0.72 | 7.3% |
| Length II | 33.7 | 2.53 | 7.5% |
| Length III | 35.1 | 2.86 | 8.1% |
| Length IV | 13.6 | 0.93 | 6.8% |
| Length V | 13.2 | 0.96 | 7.3% |
| Length VI | 9.5 | 0.78 | 8.2% |

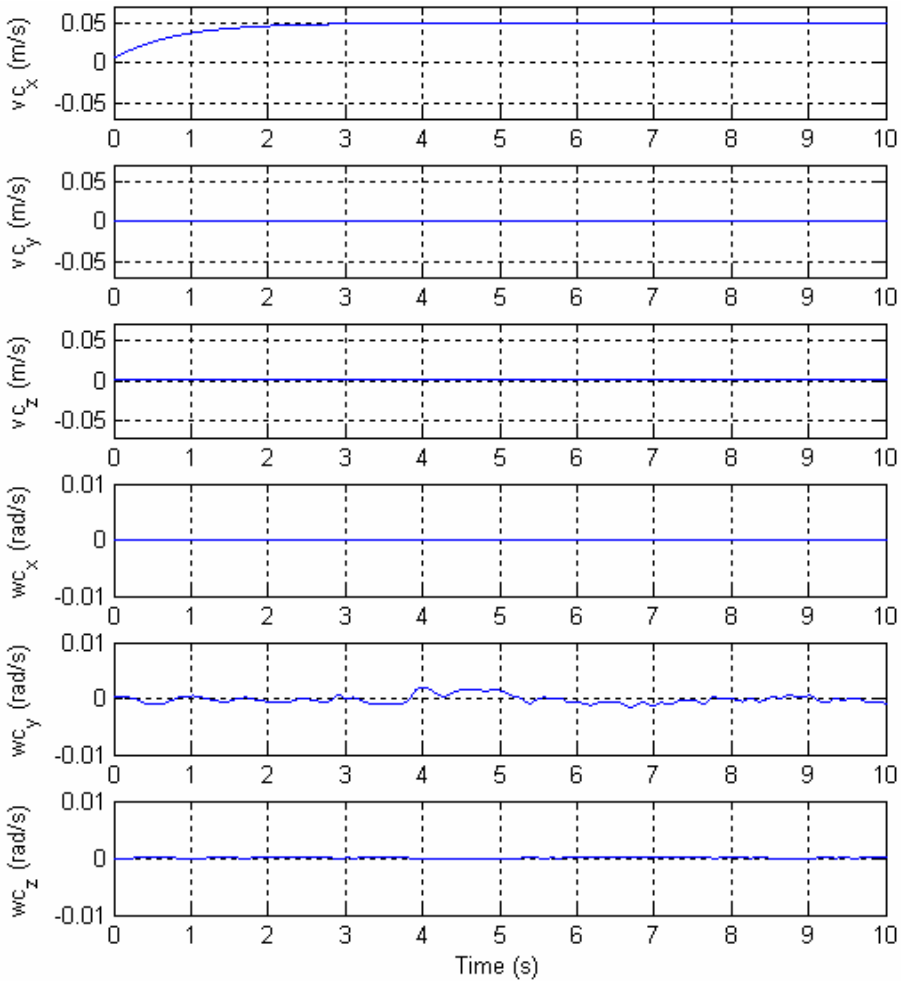


**Figure 2.13: HD camera experiment – Camera velocities for straight-line trajectory**

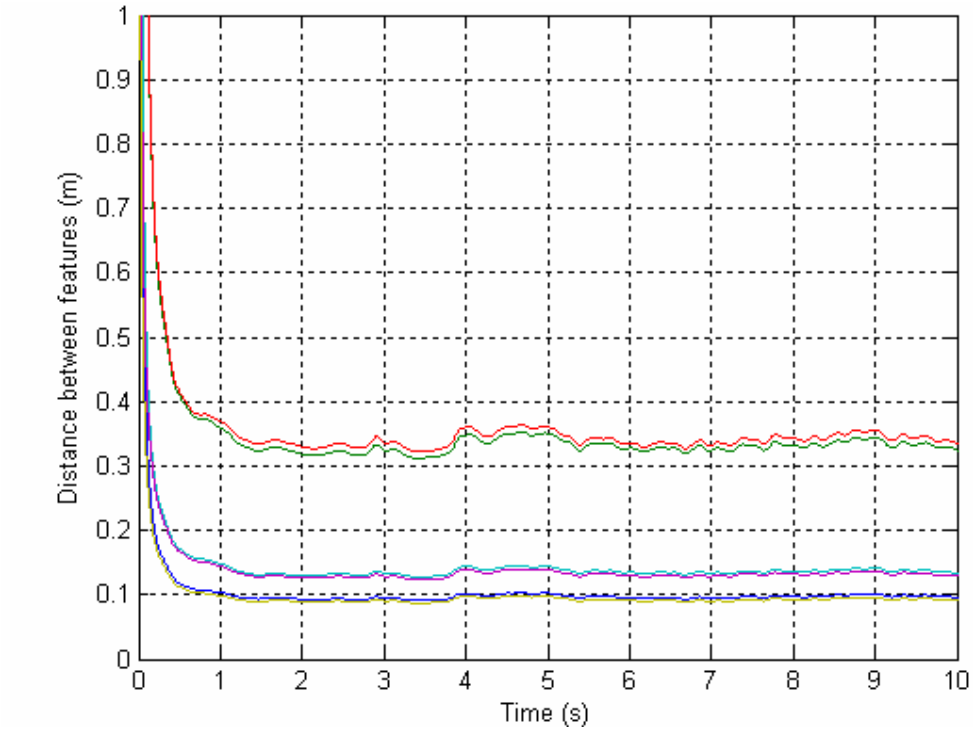


**Figure 2.14: HD camera experiment – Estimated distance between features for straight-line trajectory**

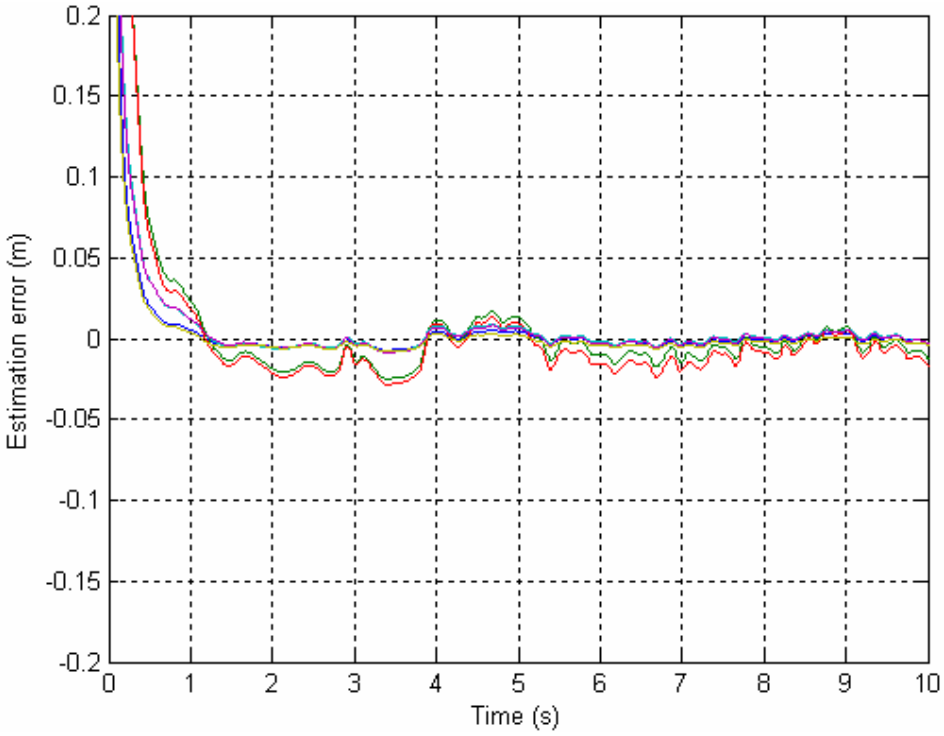


**Figure 2.15: HD camera experiment – Distance estimation error for straight-line trajectory**

Sinusoidal Trajectory

The sinusoidal trajectory selected for the mobile robot was

$$\begin{cases} x(t) = kt \\ y(t) = a\left[\cos(k\omega t) - 1\right] \end{cases}. \tag{2.33}$$

where $k$ is the velocity along the x-axis, $a$ is the amplitude of the trajectory, and $\omega$ is the angular frequency of the trajectory. As explained in Appendix C, the translational velocity $v_t$ and rotational velocity $v_\omega$ for the mobile robot's sinusoidal trajectory are defined as follows:

$$v_t = k\sqrt{a^2\omega^2 \sin^2(k\omega t) + 1} \tag{2.34}$$

$$v_\omega = \frac{-ak\omega^2 \cos(k\omega t)}{a^2\omega^2 \sin^2(k\omega t) + 1} \tag{2.35}$$

The values for $k$, $a$, and $\omega$ were

$$\begin{aligned} k &= 0.05 m/s \\ a &= 0.05 m \\ \omega &= 2\pi \ rad/s \end{aligned} \quad . \tag{2.36}$$

These parameters produce the sinusoidal trajectory shown in Figure C.1. A low-pass filter with a cutoff frequency of 0.2 Hz was applied to the feature points and camera velocities from the text file to smooth the data. The following velocity estimator gains were used:

$$K_1 = diag\{1,1\}, K_2 = diag\{1,1\}, K_3 = diag\{100,100\} \tag{2.37}$$

Figure 2.16 shows the camera velocities obtained from the robot velocities measured during the experiment. The distance between feature points is shown in Figure 2.17 and the estimation error in the distance between feature points is

shown in Figure 2.18. The maximum estimation error and percent error is listed in Table 2.4.

The sinusoidal trajectory has at least three times as much error as the straight-line trajectory. This increased error is in part due to the inaccuracies of the rotation velocities. As mentioned previously, the translation and rotation velocities are measured from the wheel encoders. For pure translations, the robot's wheels have virtual no slip and the translational velocity is quite accurate. For rotations, however, the robot's wheels must slip because there are four fixed wheels. The wheel slippage causes the rotational velocity to be less accurate compared to the translational velocity.

Additionally, the sinusoidal trajectory has more error due to the velocity estimator. The camera velocities obtained from the sinusoidal trajectory experiment were used as input to the simulation described in Appendix A. The results of the simulation are shown in Figure 2.19. To get accurate results in this simulation, the $K_3$ estimator gain had to be twenty times higher than normal. In simulation, increasing the gains has no effect except to yield better results. In an experiment, however, increasing the gains increases the amount of error in the results.

**Table 2.4: HD camera experiment – Experimental error in distance estimation for sinusoidal trajectory**

| Object | Actual Distance (cm) | Maximum Error (cm) | Percent Error |
|---|---|---|---|
| Length I | 9.8 | 2.50 | 25.5% |
| Length II | 33.7 | 7.24 | 21.5% |
| Length III | 35.1 | 7.76 | 22.1% |
| Length IV | 13.6 | 2.90 | 21.3% |
| Length V | 13.2 | 2.61 | 19.8% |
| Length VI | 39.0 | 8.61 | 22.1% |

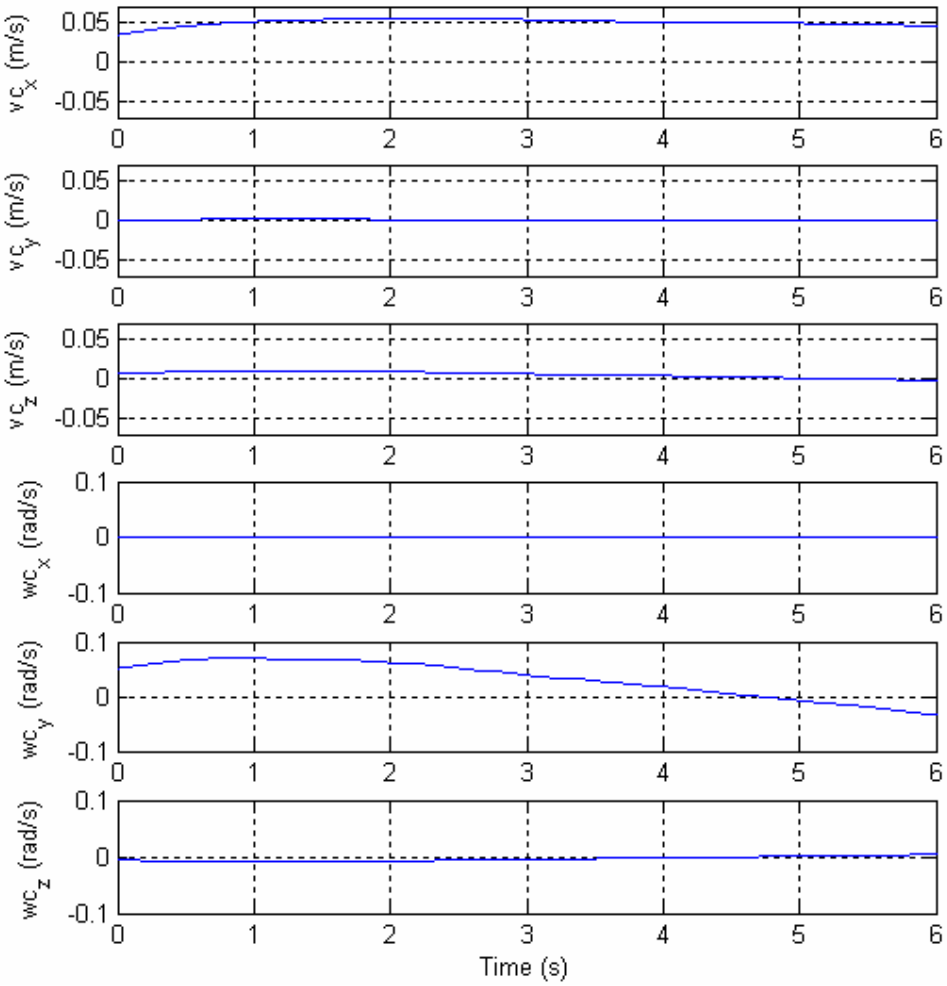


**Figure 2.16: HD camera experiment – Camera velocities for sinusoidal trajectory**

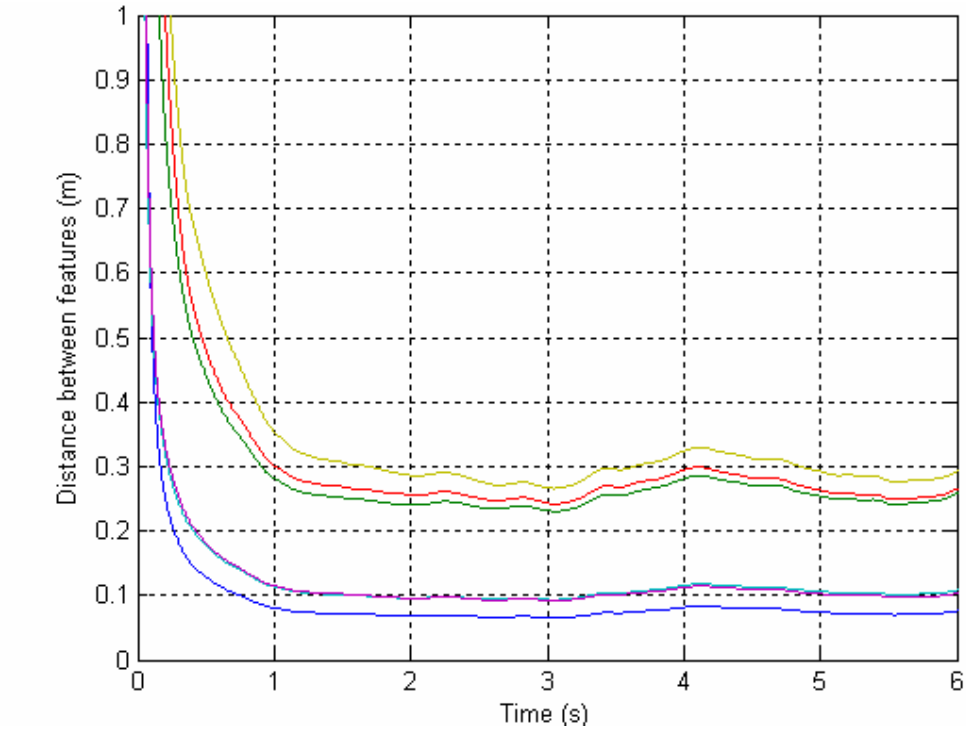


**Figure 2.17: HD camera experiment – Estimated distance between features for sinusoidal trajectory**

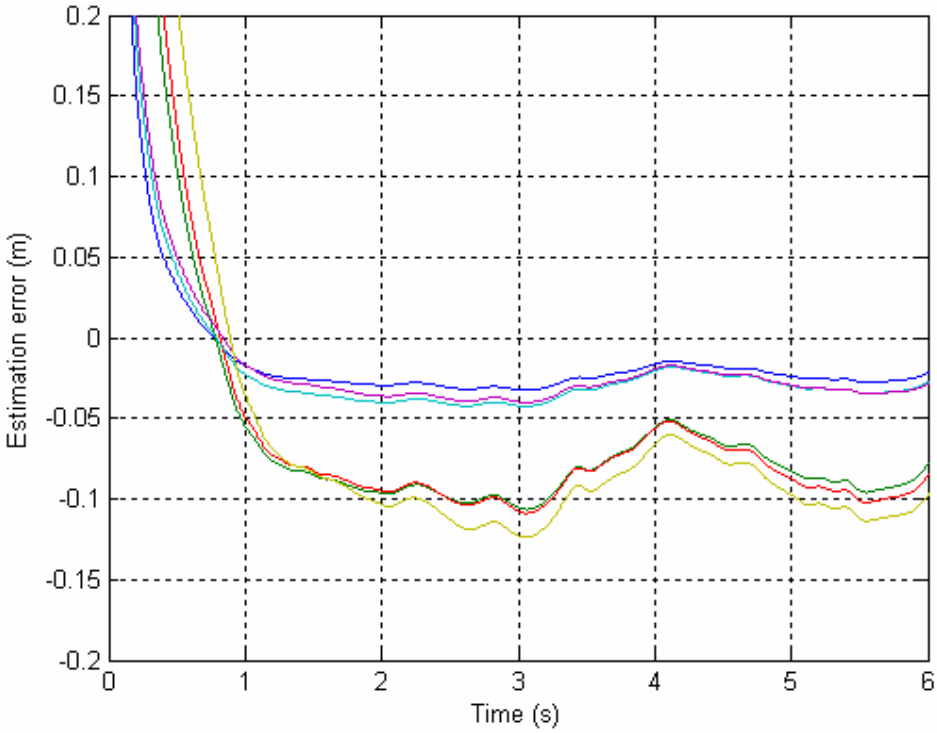


**Figure 2.18: HD camera experiment – Distance estimation error for sinusoidal trajectory**

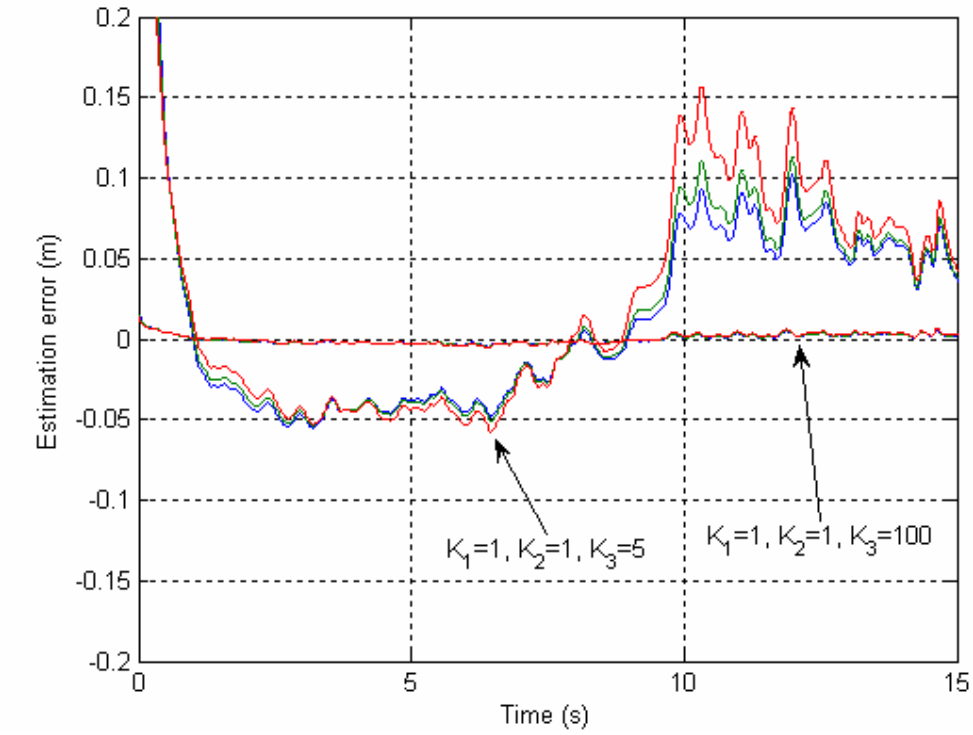


**Figure 2.19: HD camera experiment – The effect of different estimator gains on the estimation error using measured velocities and a simulated camera**

# CHAPTER 3

# DESIGN OF TEST SYSTEM FOR THE ARGON ENVIRONMENT ELECTRICAL STUDY

## 3.1 Initial Proposal

### *3.1.1 Overview*

The first step in this study was to present the customer with a research proposal that included the projected tasks, schedule, and expenses for the project. This proposal consisted entirely of estimates because the final test plan had not been written and the test system had not been designed completely. The proposal was presented to the customer to identify the project's scope, length, and cost.

### *3.1.2 Projected Tasks*

The tasks listed below were developed at the request of the customer. The team outlined these major tasks from the information given by the DOE.

1. Safety Considerations
    a. Review the safety codes and standards for high voltage systems (e.g., NFPA-70E).
    b. Speak with qualified personnel about high voltage systems and testing.
    c. Establish a safety protocol plan for the project and future laboratory testing.
    d. Write a short technical report.

2. Literature Search
    a. Gather information concerning argon and/or helium glove boxes or similar.
    b. Review open literature for electrical equipment, both dielectric and heat transfer differences, when operating in argon and/or helium environments.
    c. Investigate theory of dielectric breakdown strength of argon and helium versus air.
    d. Determine if any ANSI/IEEE standards exist for operation in inert gases.
    e. Investigate existence of any equipment manufacturer data/testing/certification of operation in argon and helium.
    f. Write a technical report on the literature search.
3. Connector Pin Arcing
    a. Procure samples of glove box feed through connectors proposed for PDCF and competing connector models.
    b. Investigate if high voltage rated connectors are available.
    c. Perform high-potential testing in feed through samples in air, argon, and helium at the Clemson Engineering Technologies Laboratory (CETL).
    d. Perform continuous load testing of feed through connectors at 480 VAC three-phase in air, argon, and helium at CETL.
    e. Write a technical report on dielectric test results.

4. Heat Capacity Derating
    a. Perform testing on electrical equipment (i.e., motors, etc) running in air and argon to determine heat capacity/transfer difference at CETL.
    b. Instrument equipment such as motors with thermocouples.
    c. Monitor with IR equipment.
    d. Monitor electrical loads and ampacity derating issues for argon.
    e. Write a technical report on heat transfer test results
5. Analytical/Empirical Modeling
    a. Develop a mathematical model for dielectric breakdown in argon and helium filled spaces.
    b. Validate the theoretical analysis using test data from Task 3.
    c. Develop a mathematical model for heat transfer in argon and helium filled spaces
    d. Validate the theoretical analysis using test data from Task 4.
    e. Write a technical report on modeling and validation activities.

### *3.1.3 Projected Schedule*

In addition to listing the tasks to be completed, the team also sought to quantify the number of months each major task would take to complete. Table 3.1 shows the list of major tasks along with an estimated number of months for each task.

**Table 3.1: Projected schedule**

| Major Task | Number of Months |
|---|---|
| Safety Considerations | 1 |
| Literature Search | 2 |
| Design of Test System | 2 |
| Connector Pin Arcing | 3 |
| Heat Capacity Derating | 3 |
| Analytic/Empirical Modeling | 2 |
| Analysis and Additional Studies | varies |
| Recommendations | 1 |
| | |
| Total | 14+ |

### *3.1.4 Projected Expenses*

To give the customer an idea of what the project might cost, a list of projected expenses was created shown in Table 3.2. Since the test plan was not finalized when this list was developed, many changes were made to this list over time.

**Table 3.2: List of projected expenses**

| | Estimated Cost |
|---|---|
| *Glovebox Supplies* | |
| Plexiglass | $ 1,000 |
| Plexiglass corner support | $ 400 |
| Electric motor | $ 200 |
| Motor mounting brackets | $ 100 |
| Motor load | $ 100 |
| Connectors | $ 200 |
| Electrical panel | $ 500 |
| Relays | $ 100 |
| Wire | $ 100 |
| Exhaust tubing | $ 100 |
| Argon | $ 300 |
| Helium | $ 300 |
| Pressure gauge | $ 50 |
| | |
| *Test Equipment* | |
| Computer w/ data acquisition and control board | $ 2,500 |
| IR equipment for temperature measurements | $ 800 |
| Thermocouples | $ 200 |
| Voltage/current meters for motors | $ 1,000 |
| High-speed video camera to record arcing* | $ 2,000 |
| | |
| *Outsourcing Costs* | |
| Machine shop time | $ 600 |
| Inspection of electrical connectors* | $ 600 |
| | |
| *Other* | |
| Misc. supplies and materials | $ 500 |
| Rubber gloves | $ 40 |
| | |
| **Total projected expenses** | $ 11,690 |

The items marked with an asterisk (*) were found to be unnecessary. The total of the projected expenses without these items is $9,090. One of the costs not included on this lists is the cost of the hydraulics.

## 3.2 Safety Review

### *3.2.1 Overview*

This study involves some electric motors operating at 480 VAC three-phase. However, the Tritium Extraction Facility (TEF) at the Savannah River Site (SRS) discovered that in some cases the variable-frequency drives (VFDs) for these motors had a peak voltage of 1,000V with reference to ground. Additionally, the peak voltage between two phases could be as much as 1,000 VAC. For this reason, the safety requirements for operation at 1,000 VAC phase to phase should be followed.

### *3.2.2 Approach Distance*

According to the 2002 National Electric Safety Code (NESC) Handbook, employees must maintain an approach distance from energized conductors or parts. This approach distance depends on the operating voltage. For 1,000 VAC, the approach distance is 0.67 m or 2.2 ft. The NESC Handbook states that "supply employees must not approach energized parts or take conductive objects near energized parts" within this approach distance without meeting the one of the requirements listed below from the handbook [27].

- The line or part is de-energized and grounded.
- The employee is insulated from the energized line or part using insulated tools, gloves, rubber gloves, or rubber gloves with sleeves.
- The energized line or part is insulated from the employee and any other line or part at a different voltage.

In addition, the "precautions for approaching voltages from 301 V to 72.5 kV" listed below from the handbook must be followed [27].

- Employees must be protected from phase to phase and phase to ground differences in potential.
- Exposed grounded lines, conductors, or parts must be guarded or insulated.
- When the rubber glove method is used, the gloves must be insulated for the maximum use voltage in NESC Table 441-6.
- When the rubber glove method is used, it must be used with one of the following two following methods:
  - Rubber insulating sleeves which are insulated for the maximum use voltage in NESC Table 441-6.
  - Insulating exposed energized lines or parts within the employee's maximum reach (this does not apply to the part being worked on).

### *3.2.3 Equipment Ratings*

Relays, wires, and equipment which are not being tested should be rated at 1,000 VAC or higher. This requirement will protect employees and equipment from the operating voltage which is suspected to be greater than 480 VAC. Any equipment that is being tested should be monitored either visually or with automatic shutoffs to prevent damage to the equipment.

## 3.3 Literature Review

The lower dielectric strength of argon compared to air presents a problem for high-voltage systems. A number of works have been written on the breakdown voltage in gases. Meeks and Craggs provide one of the most comprehensive and up-to-date works on this subject [28]. The authors discuss the breakdown voltage for argon and introduce an approximate formula for this voltage. A number of factors determine the point at which electrical breakdown occurs in a gas. These factors include the magnitude and frequency of the voltage, the spacing and type of electrodes, and both the pressure and temperature of the gas (i.e. gas density).

Paschen's law is an important principle for the electrical breakdown of gases. It states that the product *pd* of the pressure *p* and distance between the electrodes *d* is constant for a particular breakdown voltage. To see the impact of various factors that determine the breakdown of argon, the breakdown voltage for air vs. argon at low pressures and high frequencies has been investigated [29]. A phenomenon related to electrical breakdown in gases is partial discharge within insulators. This factor is important because the voltage at which partial breakdown occurs can be lower than the breakdown voltage for air [30]. Finally, because arcing is a serious issue in the petroleum industry, this industry has presented standards to eliminate arcing or to reduce its effects [31].

In addition to the electrical breakdown characteristics of argon, the thermal properties of argon gases have also been studied. A theoretical method for calculating thermal and electrical properties of argon has been formulated [32]. Experimental data for the thermal properties of air and argon has also been collected [33].

In summary, the majority of the work currently available is theoretical in nature. This literature is helpful when developing a model, but does not pertain to this specific case of operating electric motors and electrical connectors in an argon environment. Testing is necessary to determine the breakdown voltages for particular connectors and motors and to investigate heat transfer issues for particular electric motors.

## 3.4 Test Plan

### *3.4.1 Overview*

This section outlines the plan for the investigation of connector pin arcing, heat capacity derating, and motor arcing. There are essentially two separate tasks. The connector pin arcing task will use a variable-frequency drive (VFD) with resistive load to test the connector. The heat capacity derating and motor arcing task will bypass the connectors with direct wiring and will use hydraulics to load the motors.

### *3.4.2 Connector Pin Arcing*

The following procedure will be used to investigate the connector pin arcing. The purpose is to determine whether the proposed connectors (see Appendix D) will experience any arcing issues when operating in an argon environment.

1. Construct a sealed Plexiglas box with T-slotted aluminum corner supports that has the ability to contain air, argon, or helium at a specific pressure and purity level.
2. Create arcing by placing two electrodes near each other and apply a voltage with a variable-frequency drive (VFD). A resistive load may be attached to the electrodes to determine the current. Perform this test separately for air, argon, and helium in the sealed box. Record the magnitude and frequency of the applied voltage, temperature, pressure, type of electrodes, and spacing of electrodes. Measure the voltage and current with arcing and compare to the voltage and current without arcing.

3. Test each of the proposed connectors individually inside the sealed box. With the following method, conduct the testing first for air, argon, and helium:
    a. Wire the connector to be tested to the VFD and a resistive load. Attach a measuring device to record the current and voltage of the VFD over time.
    b. Operate the VFD at a constant voltage and current. The voltage and current should be close to the expected values for one of the proposed electric motors. Run the VFD with the proposed connector and resistive load for four hours maintaining a constant pressure and temperature as much as possible.
    c. Observe the current and voltage for the VFD and note whether any arcing may have occurred during the test. Open the box, open the connector, and visually inspect for pitting, charred surfaces, discoloration, or any other signs indicating that arcing occurred. With a digital camera, take photographs of the connector pins.

### *3.4.3 Heat Capacity Derating and Motor Arcing*

The procedure below will be use to investigate the heat capacity derating and motor arcing. The purpose is to determine whether the proposed electric motors (see appendix D) will have any heating or arcing problems when operating in an argon environment.

1. Construct an appropriately-sized sealed Plexiglas box with T-slotted aluminum corner supports that has the ability to contain argon at a specific

pressure and purity level. The box should accommodate the largest two electric motors and two hydraulic motors. Attach thermocouples inside the box to record the temperature over time.

2. Test each of the proposed electrical motors (2 at a time) inside the sealed box. With the following method, conduct the testing first for air and then for argon:
    a. Mount two of the proposed electric motors inside the box. Couple the two hydraulic motors to the electric motors. Connect measuring devices to the motor controllers to record the current and voltage over time.
    b. Operate the electric motors for 8 hours. The hydraulic motors should be adjusted to give the electric motors a typical load.
    c. Observe the current and voltage for the motor controllers and note whether any arcing may have occurred during the test. Record the temperature increase during the testing. Compare the rise in temperature for air versus the argon environment.

## 3.5 System Design

### *3.5.1 Philosophy*

The design of the test system emphasizes safety and reduced costs. For safety, the systems features electrical lockout. In addition, a computer controls the motor controllers and measures current and voltages for each motor so that human contact with the high voltage is eliminated. To reduce costs, the Plexiglas box

contains only the 2 electric motors to be tested and 2 hydraulic motors for a load. By doing so, the box can be smaller and less expensive.

### *3.5.2 Connector Pin Arcing*

The systems diagram for the connector pin arcing tests is shown in Figure 3.1. The components of this system will be explained in this section.

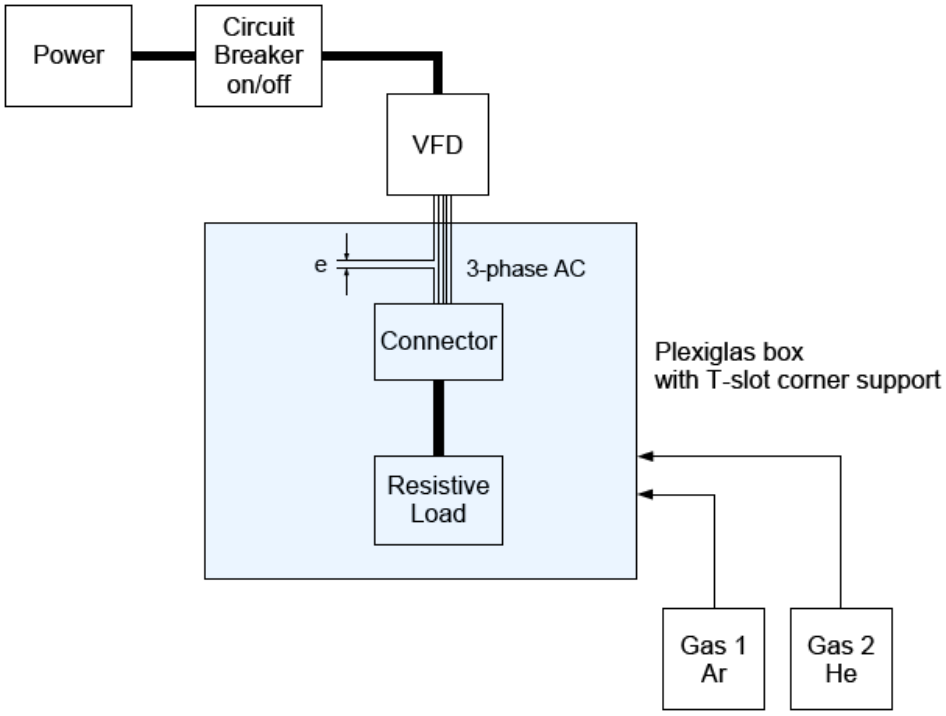


**Figure 3.1: System diagram for connector pin arcing tests**

Power

The power for the test system will be 480VAC, three-phase.

Circuit Breaker

The circuit breaker will feature will feature an electrical lockout. This feature prevents the power from being switched on while someone is working with the equipment.

VFD

The variable-frequency drive (VFD) is used to control the voltage and frequency applied to the connector. Input to the VFD is 480VAC, three-phase.

Plexiglas Box

The Plexiglas box has T-slotted aluminum corner supports and houses the electric motors and the hydraulics inside an air or argon environment. The same box used for the heat capacity derating and motor arcing test may be used or a separate box may be constructed. The dimensions of the box should be at least 1x1x0.5 ft.

Electrodes

Two electrodes are placed at a distance *e* from each other to investigate what happens when arcing occurs.

Connectors

The connectors to be tested will be placed inside the box. The proposed connectors are shown in Appendix D.

Resistive Load

The resistive load is attached to the connector to determine the amount of current supplied by the VFD. Ideally, the current would be equal to the current for a typical 480VAC three-phase electric motor.

### *3.5.3 Heat Capacity Derating and Motor Arcing*

The system diagram for the heat capacity derating and motor arcing tests is shown in Figure 3.2. The components of this system will be explained in this section.

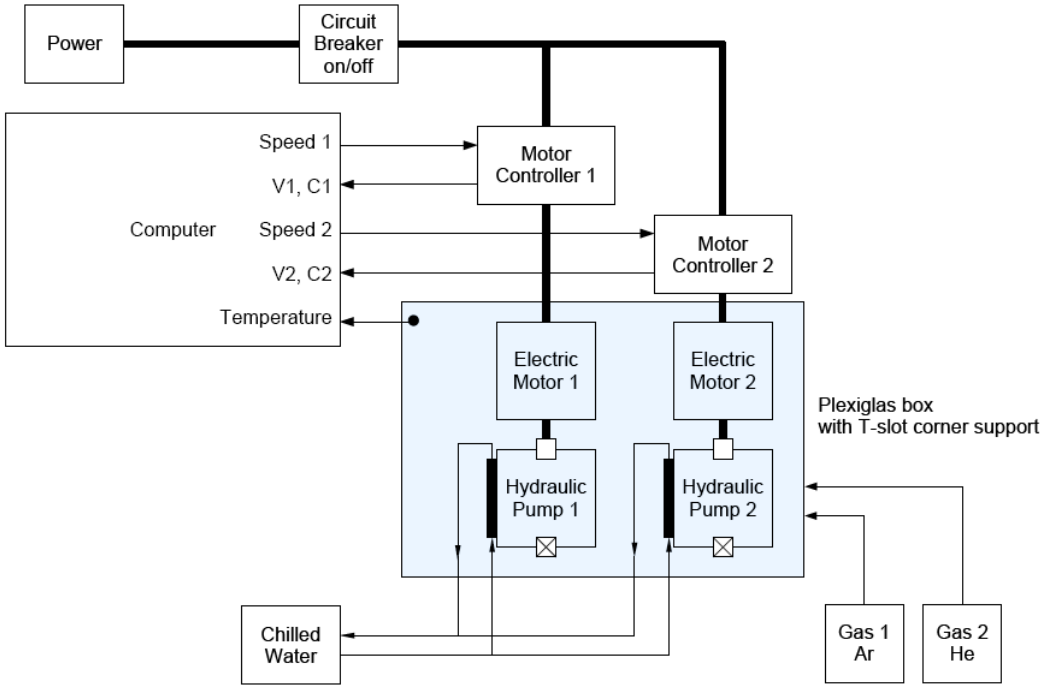


**Figure 3.2: System diagram for heat capacity derating and motor arcing tests**

Power

The power for the test system will be either 120VAC or three-phase power (240V or 480V).

Circuit Breaker

The circuit breaker will feature will feature an electrical lockout. This feature prevents the power from being switched on while someone is working with the equipment.

Plexiglas Box

The Plexiglas box houses the electric motors and the hydraulics inside an air or argon environment. The approximate dimensions of the box are 3x3x3 ft.

Computer

The computer controls the motor controller via serial ports. Additionally, the computer has data acquisition and control to measure the current and voltage applied to each motor and to measure the temperature of the box.

Motor Controllers

The motor controllers vary depending on the motor. Some motors only have a power supply with a speed adjustment knob. Other motors have a motor controller built in. For the induction motors, the motor controller is a VFD.

Electric Motors

Two motors will be tested at a time. These motors are connected directly to the motor controller (none of the proposed connectors included). A list of motors with pictures is shown in Appendix D.

Hydraulic Pumps

Hydraulics are used to load the motors. If needed, the hydraulic motors will be cooled with chilled water.

# CHAPTER 4

# CONCLUSIONS

## 4.1 Euclidean Position Estimation of Static Features using a Moving Camera with Known Velocities

In this work, we have presented an estimation strategy for 3D Euclidean reconstruction of static features on an object using a single moving camera whose velocities are known. The proposed estimator has a simple mathematical structure and can be easily implemented. Numerical simulations and experimental results using a mobile robot in an indoor environment were presented. These results demonstrate that the estimation strategy is accurate and converges quickly, in under one second, even with a poor resolution and low frame rate camera. With a high-definition camera, the results are slightly improved. Further experimental validation using a video camera mounted on a UAV is being considered. For future work, the real-time performance of this algorithm may be evaluated.

## 4.2 Design of Test System for the Argon Environment Electrical Study

The process of design starts with a goal. In this case, the goal was to investigate the problems associated with operating electrical connectors and electric motors in an argon environment. The next step is researching possible options to meet the goal. This step involves becoming familiar with the theory associated with the subject and researching what has already been discovered in the field. From this information, a plan must be created to meet that goal. The plan formulated to meet the goal of this study was outlined in the previous chapter. The final step is

verification and testing. The plan is put into action and tested to determine whether it meets the goal.

Design is an iterative process. The designer must work with the customer to identify needs and work to meet those needs. The designer must have a clear understanding of what the customer expects in order to satisfy the customer. Often the initial design must be changed when more information becomes available.

The next step is to construct the test system and perform the tests that have been outlined. Many future studies can be conducted with this test system. The additional tests that would be valuable depend on the initial test results. For instance, if the connectors do not experience any arcing, the voltage could be increased to determine the point at which the connectors do experience arcing. On the other hand, if the connectors do experience arcing, additional tests can be conducted to determine how the potential for arcing can be reduced. Possible options for reducing the tendency for arcing would be to increase the pressure, use connectors with a high voltage rating, or investigate a nitrogen environment. Furthermore, if heat transfer in an argon environment is a problem, additional testing would provide a basis for recommendations to work around the problem. For example, tests could be conducted using a fan and/or a cooling system. Finally, the customer noted that the performance of wireless instrumentation in an argon environment might need to be evaluated in the future. Testing of wireless instrumentation would require further changes to the design.

# APPENDICES

# Appendix A

# Simulink Model

## A.1 Overview

### *A.1.1 Background*

The Simulink model shown in Figure A.1 consists of the following major subsystems:

- Camera Kinematics
- Image Velocity Estimator
- Depth and Euclidean Points Calculation
- Error Subsystem

The Camera Kinematics block simulates a moving camera. The output of this block is the camera properties and the image points (tracked feature points). The Image Velocity Estimator block estimates the image velocities using the estimator described previously. The Depth and Euclidean Points Calculation uses the camera properties and image velocities to find the depth and Euclidean feature points. Finally, the Error Subsystem determines the error in depth estimation.

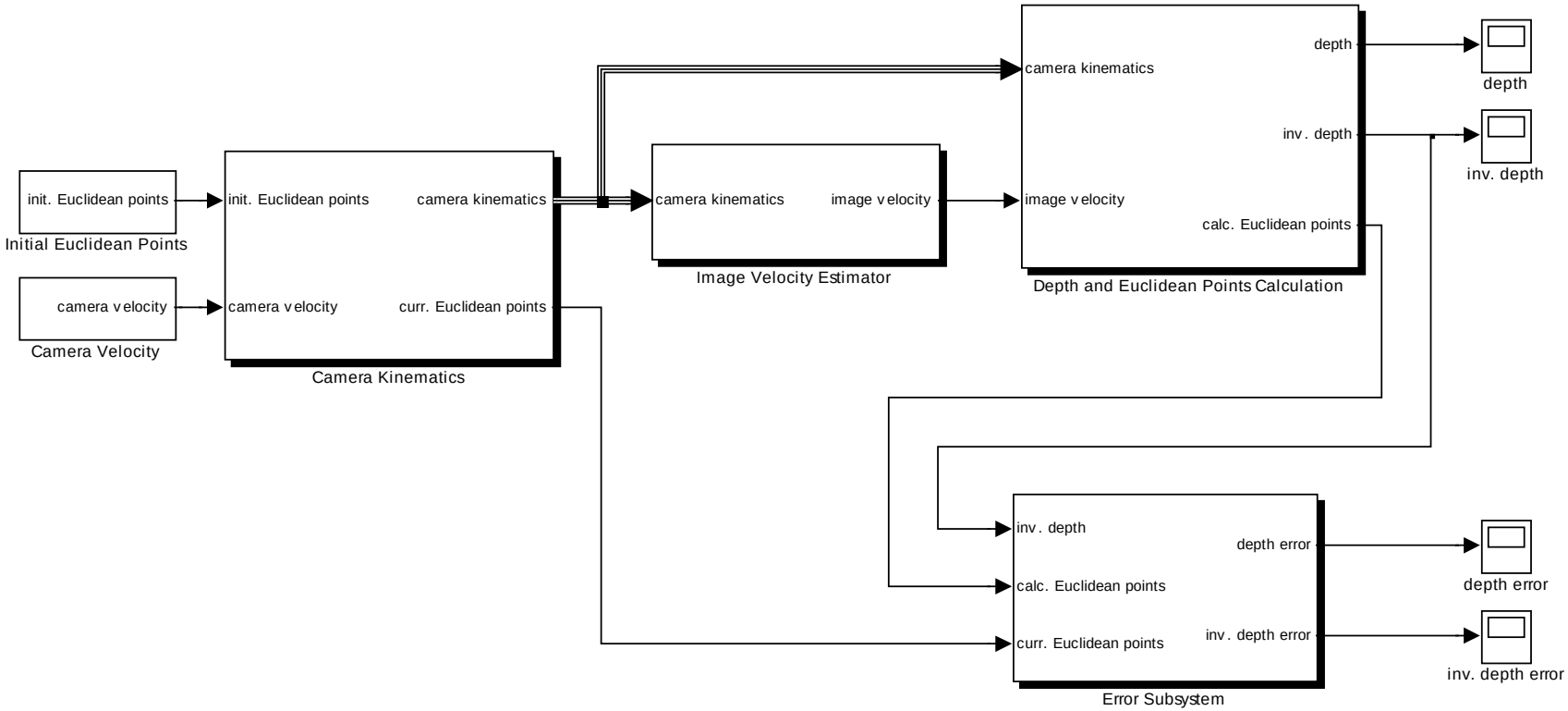


**Figure A.1: Structure from motion model**

## A.2 Camera Kinematics

### *A.2.1 Overview*

The Camera Kinematics subsystem simulates the camera and feature tracker. It calculates the position of the feature points in the image plane for a moving camera. The inputs are the camera velocity (both translation and rotation) and the initial Euclidean feature points in the camera frame. The subsystem calculates the new Euclidean feature points with respect to the camera and uses that calculation to find the new image points. The block has the ability to simulate white Gaussian noise. This feature can be turned on using the parameters dialog box. The block also can round off the image points to integers to simulate feature trackers without sub-pixel accuracy. The low-pass filter subsystem in the block simulates what would happen if a low-pass filter were used to reduce the noise.

### *A.2.2 Camera Velocity Input*

The camera velocity input to the Camera Kinematics block is the following $6\times1$ vector:

$$v = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \tag{A.1}$$

The camera's translation velocity is $v_c \in \mathbb{R}^3$ and the rotation velocity is $\omega_c \in \mathbb{R}^3$. The block uses embedded MATLAB code to generate the camera velocity as shown in Figure A.2 and Figure A.3.

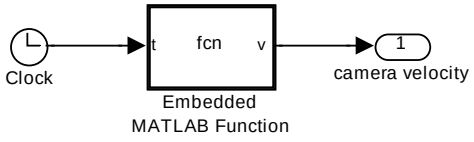


**Figure A.2: Camera Velocity**

```
function v = fcn(t)
% Create velocity vector

v = [0.2*cos(t); 0.2*sin(t); 0.1*cos(t); 0; 0; 0.1*sin(2*pi*0.1*t)];
```

**Figure A.3: Embedded MATLAB code within Camera Velocity**

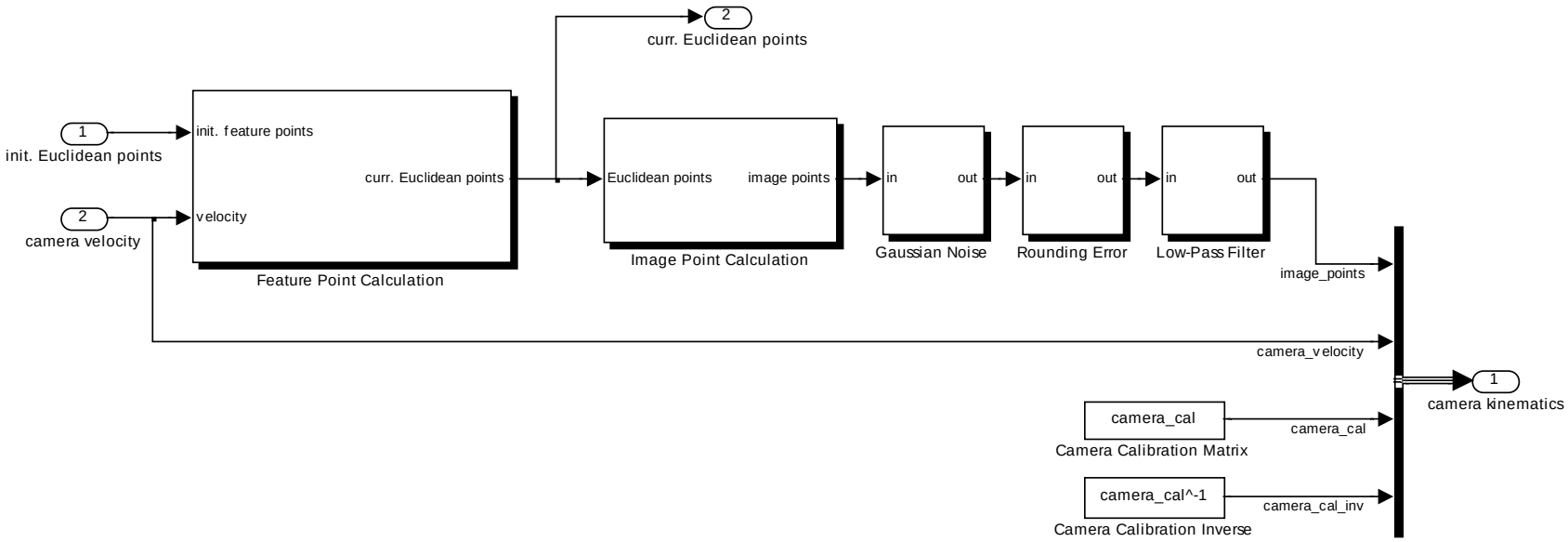


Figure A.4: Camera Kinematics

### *A.2.3 Initial Euclidean Points Input*

The initial Euclidean feature points come from a $3 \times n$ vector of features, where $n$ is the number of features. These feature points in the camera's reference frame.

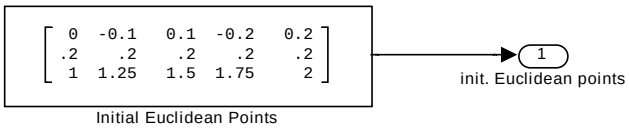


**Figure A.5: Initial Euclidean Points**

### *A.2.4 Parameters*

The parameters dialog box sets the camera calibration matrix and allows the user to turn on or off the white Gaussian noise, rounding of image points to integers, and the low-pass filter. The noise variance and the cutoff frequency of the low-pass filter can be set using this dialog box.

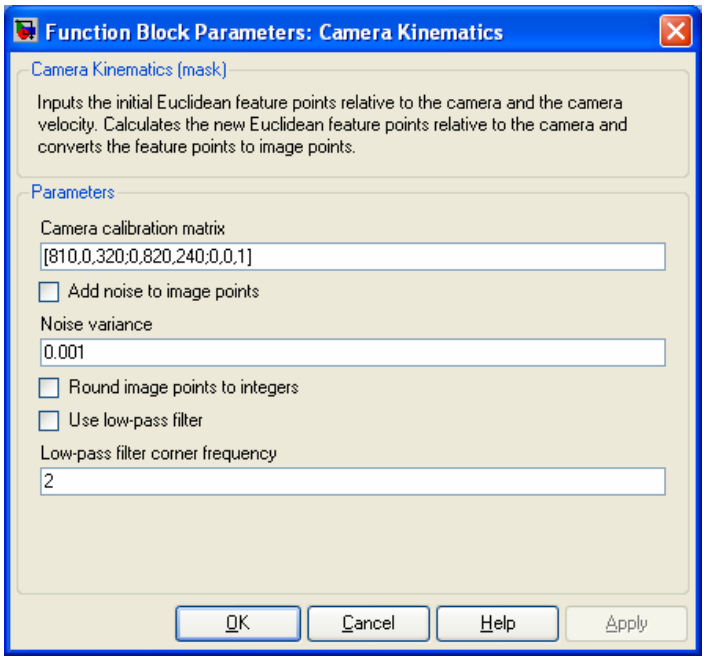


**Figure A.6: Initial Euclidean Points parameters dialog box**

**Table A.1: Camera Kinematics parameters**

| Prompt | Variable |
|---|---|
| Camera calibration matrix | *camera_cal* |
| Add noise to image points | *add_noise* |
| Noise variance | *noise_variance* |
| Round image points to integers | *add_rounding* |
| Use low-pass filter | *filter_noise* |
| Low-pass filter corner frequency | *fo* |

### *A.2.5 Feature Point Calculation*

The Feature Point Calculation block finds the new 3D Euclidean position of the feature points. The velocity of the feature points with respect to the camera is given by

$$\dot{\bar{m}}_i = -v_c - \omega_c \times \bar{m}_i \tag{A.2}$$

where $v_c$ and $\omega_c$ are the camera translation and rotation velocities, and $\bar{m}_i$ is the 3D Euclidean position of the $i^{th}$ feature point with respect to the camera frame $\mathcal{I}$. The block uses an embedded MATLAB function to compute the velocity and an integrator to find the position.

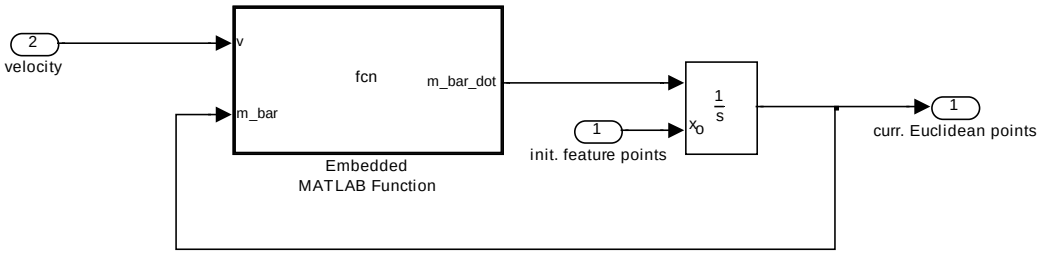


**Figure A.7: Feature Point Calculation**

```
function m_bar_dot = fcn(v, m_bar)

n = size(m_bar,2);
m_bar_dot = zeros(3, n);
vc = v(1:3,1);
wc = v(4:6,1);

for i=1:n
    m_bar_dot(:,i) = -vc - cross(wc, m_bar(:,i));
end
```

**Figure A.8: Embedded MATLAB code within Feature Point Calculation**

### *A.2.6 Image Point Calculation*

The Image Point Calculation subsystem uses the following equation derived from Eq. (2.2) and Eq. (2.4) to find the image points:

$$p_i = A \cdot \frac{\overline{m}_i}{z_i} \tag{A.3}$$

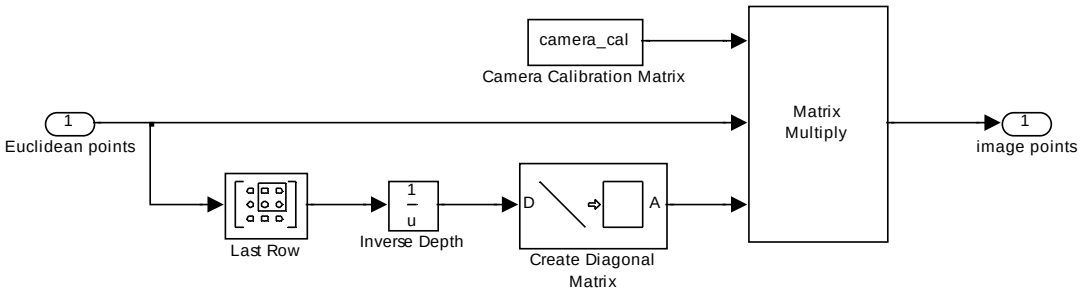


**Figure A.9: Image Point Calculation**

### *A.2.7 Gaussian Noise*

The Gaussian Noise block adds a different random number to the *x* and *y* values of the image points. The variance of the random number is determined by the *noise_variance* parameter. If the parameter *add_noise* is false, the signal passes through the block without any change.

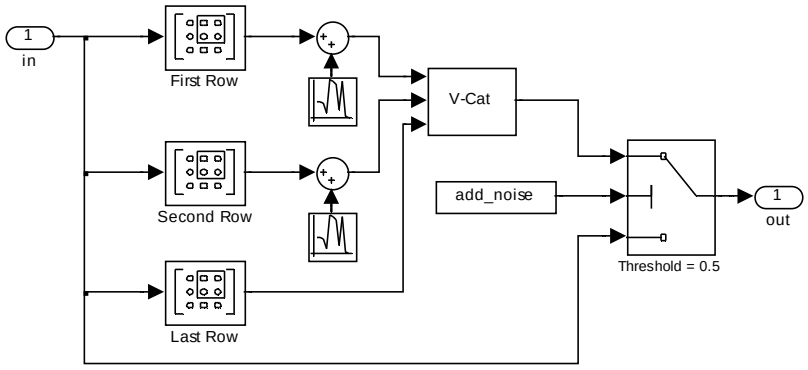


**Figure A.10: Gaussian Noise**

### *A.2.8 Rounding Error*

The Rounding Error subsystem converts the image points to integers. The parameter *add_rounding* must be checked for the block to be active.

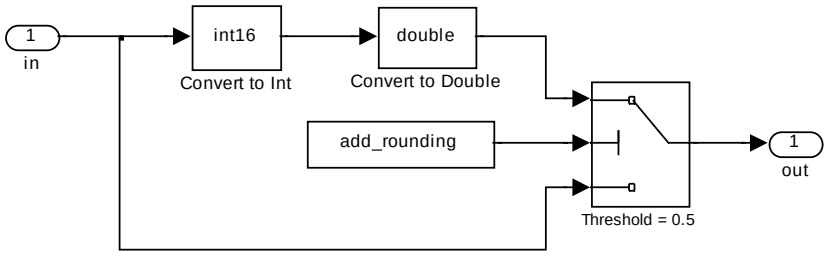


**Figure A.11: Rounding Error**

### *A.2.9 Low-pass Filter*

The Low-pass Filter block implements a first-order low-pass filter with a corner frequency of $f_o$ (in Hz). The transfer function is given by

$$H(s) = \frac{2\pi f_o}{s + 2\pi f_o}. \qquad \text{(A.4)}$$

The low-pass filter can be turned off if *filter_noise* is false.

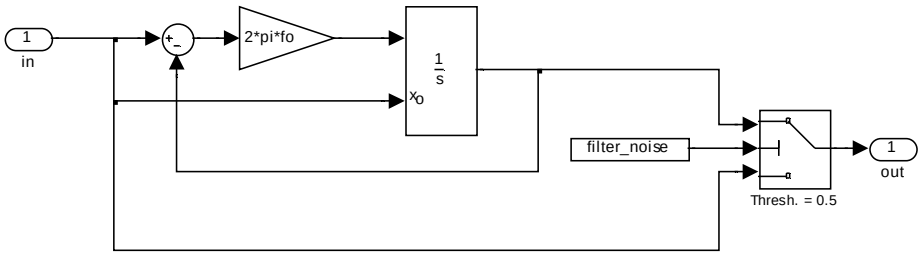


**Figure A.12: Low-pass Filter**

## A.3 Image Velocity Estimator

The Image Velocity Estimator subsystem computes the velocities of the tracked image points. Within this system, the Pixel Displacement block computes the difference between the original pixel location and the current pixel location. The Velocity Estimator block estimates the derivate of the result. This block also has the ability to replace the Velocity Estimator with a simple derivative operator by changing a parameter.

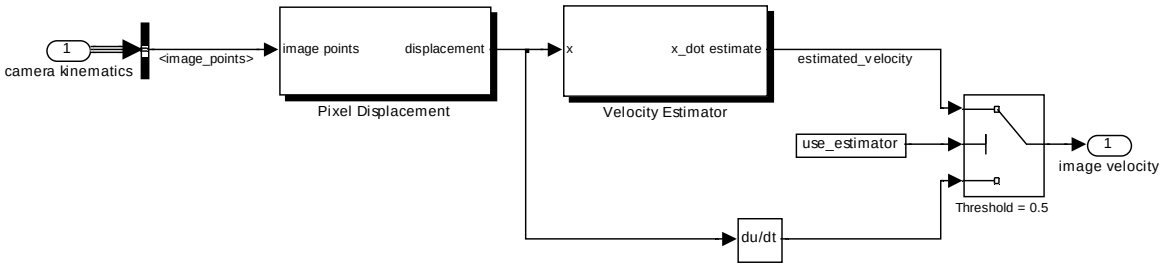


**Figure A.13: Image Velocity Estimator**

### *A.3.1 Parameters*

The parameters $K_1$, $K_2$, and $K_3$ for the Velocity Estimator block can be set with the dialog box shown in Figure A.14. If "Use estimator" is checked the Velocity Estimator block is used to find the image velocities; otherwise a simple derivative operator is used.

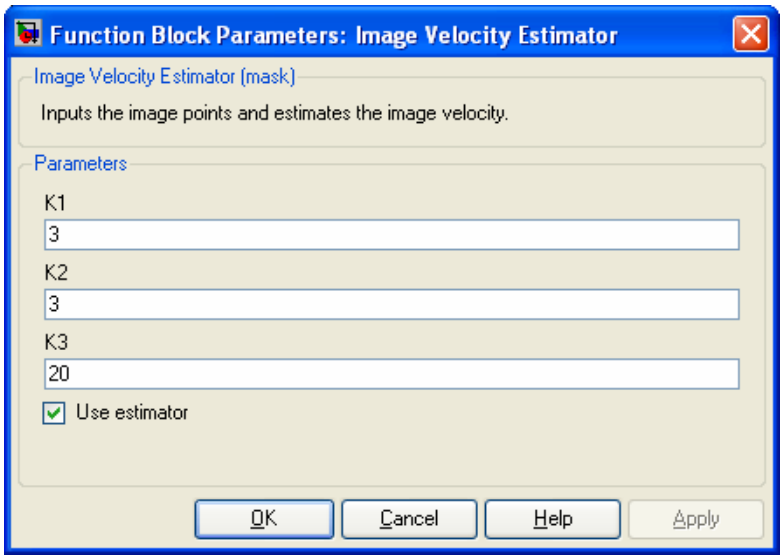


**Figure A.14: Image Velocity Estimator parameters**

### *A.3.2 Pixel Displacement*

The Pixel Displacement block subtracts the initial image points from the current image points to find the pixel displacement.

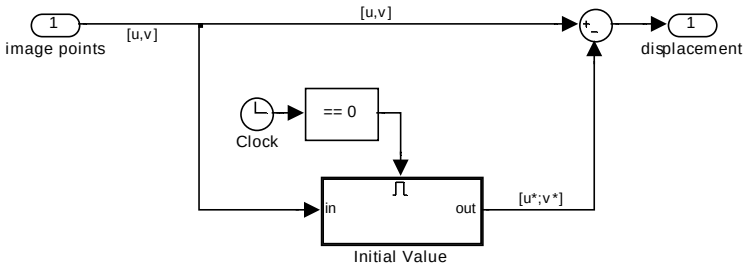


**Figure A.15: Pixel Displacement**

### *A.3.3 Velocity Estimator*

The Velocity Estimator block implements the nonlinear estimator in Eq. (2.12). The constants $K_1$, $K_2$, and $K_3$ are determined by the inputs in the Image Velocity Estimator parameters dialog box.

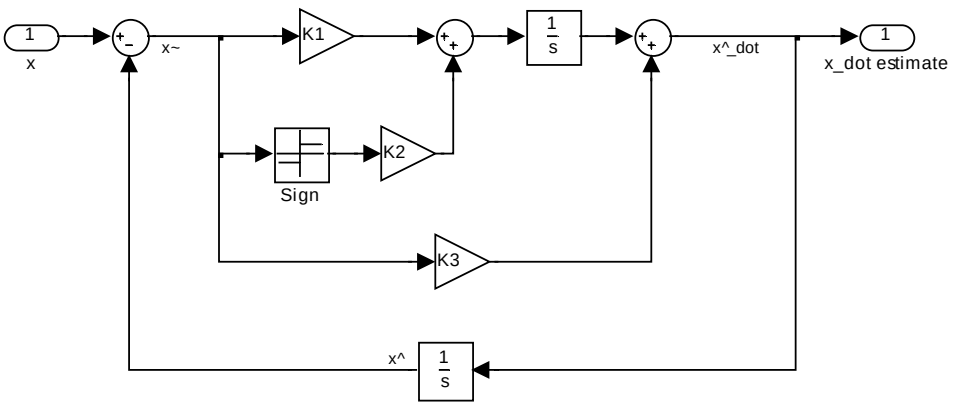


**Figure A.16: Velocity Estimator**

## A.4 Depth and Euclidean Points Calculation

This subsystem estimates the inverse depth, the depth, and the 3D Euclidean points for all feature points. The Depth Estimator produces the inverse depth estimates (and thus the depth estimates). The Euclidean Points Calculation uses the inverse camera calibration matrix and the depth to find the 3D Euclidean points.

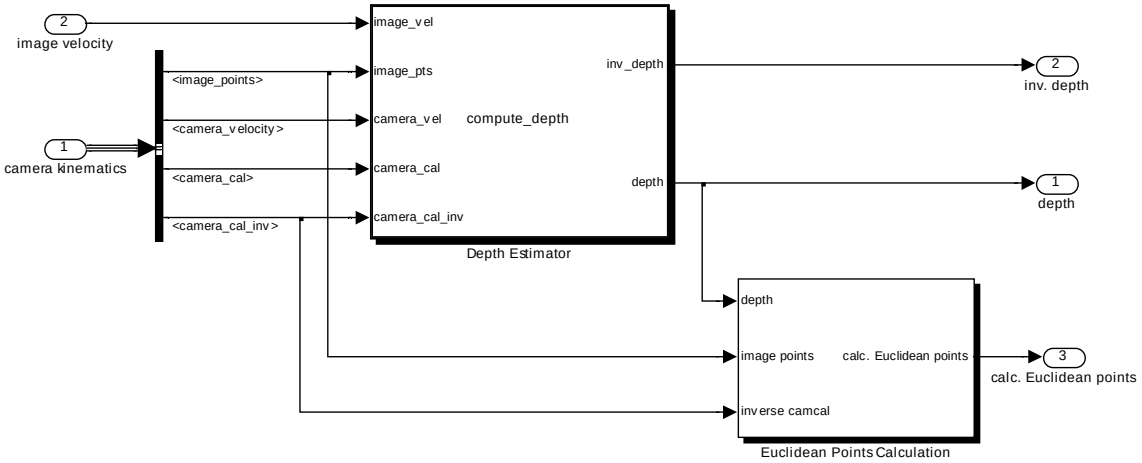


**Figure A.17: Depth and Euclidean Points Calculation**

### *A.4.1 Depth Estimator*

The Depth Estimator uses Eq. (2.24) to estimate the inverse depth (and thus the depth). The embedded MATLAB code in Figure A.18 implements this formula for each feature point.

```
function [inv_depth, depth]  = compute_depth(image_vel, image_pts,
camera_vel, camera_cal, camera_cal_inv)

% Find the number of feature points
n = size(image_pts,2);

% Initialize variables
depth = zeros(1,n);
inv_depth = zeros(1,n);
zeta = zeros(2,1);
delta = zeros(2,1);

% Find normalized image points
image_pts_norm = camera_cal_inv * image_pts;

% translation and rotation camera velocities
vc = camera_vel(1:3);
wc = camera_vel(4:6);

for i=1:n

    % Calculate extended camera calibration matrix
    x_camcal = camera_cal;
    x_camcal(1:2,3) = x_camcal(1:2,3) - image_pts(1:2,i);

    pi_ = x_camcal(1:2,:);

    % Skew-symmetric matrix
    sk_mi = [0, -image_pts_norm(3,i), image_pts_norm(2,i);
            image_pts_norm(3,i), 0, -image_pts_norm(1,i);
            -image_pts_norm(2,i), image_pts_norm(1,i), 0];

    zeta = pi_ * vc;
    delta = pi_ * sk_mi * wc;

    % inverse depth formula
    inv_depth(i) = 1 / (zeta' * zeta) * zeta' * (delta - image_vel(1:2,i));

    depth(i) = 1/inv_depth(i);

end
```

**Figure A.18: Embedded MATLAB code within Depth and Euclidean Points Calculation**

### *A.4.2 Euclidean Points Calculation*

To find the estimated 3D Euclidean position, solve Eq. (A.3) for $\overline{m}_i$ to get

$$\overline{m}_i = A^{-1} p_i z_i . \tag{A.5}$$

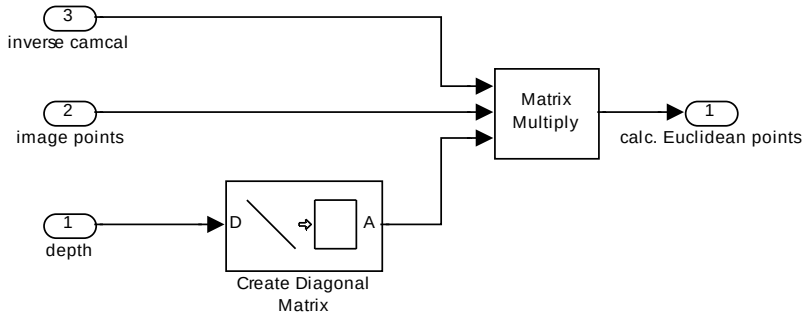


**Figure A.19: Euclidean Points Calculation**

## A.5 Error Subsystem

The Error Subsystem block finds depth error and the inverse depth error. The subsystem has a separate input for the inverse depth because the depth is limited to 10 meters. If the estimated depth is greater than 10 meters, the inverse depth will be the actual estimated inverse depth.

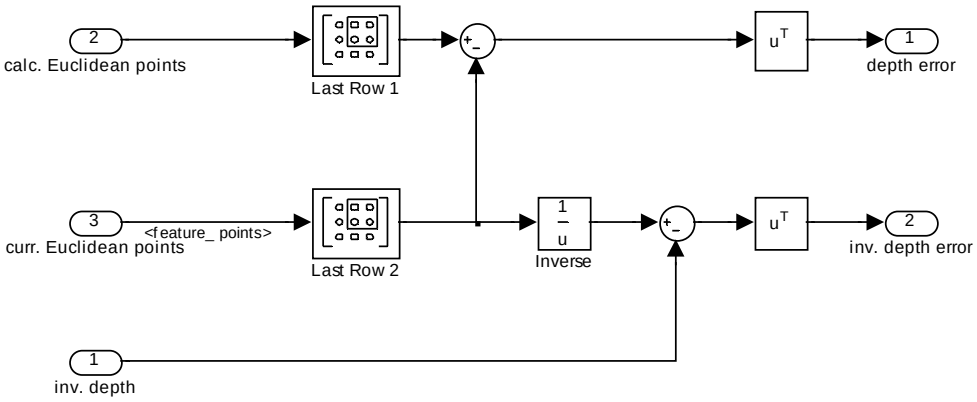


**Figure A.20: Error Subsystem**

Appendix B

Camera Calibration

## B.1 Overview

This appendix explains how the intrinsic and extrinsic calibration parameters for the camera were found using the Camera Calibration Toolbox for MATLAB [26]. The intrinsic calibration used images of a known checkerboard. The extrinsic calibration uses the intrinsic calibration parameters along with another image of the checkerboard. From the extrinsic parameters, the transformation from the robot frame to the camera frame can be found.

## B.2 Intrinsic Calibration

To find the intrinsic calibration parameters, the Camera Calibration Toolbox for MATLAB was used. The process involves using a checkerboard pattern of a known size (28mm squares in this case). At least fifteen images of the checkerboard pattern are taken from different positions and orientations. These images allow the toolbox to find the calibration parameters.

## B.3 Extrinsic Calibration

Because the velocities measured by the robot are with respect to the robot reference frame, a transformation must be found between the robot reference frame and the camera reference frame. In the diagram below, the robot reference frame is $o_0$ and the camera reference frame is $o_1$. To find the transformation, the robot is placed in a known position and orientation. Using the Camera

Calibration Toolbox for MATLAB, a transformation can be found between the camera reference frame, $o_1$, and a fixed reference frame, $o_2$, in the image. Since the transformation from the robot reference frame, $o_0$, to the fixed reference frame, $o_2$, is already known, the transformation from the robot reference frame, $o_0$, to the camera reference frame, $o_1$, can be found.

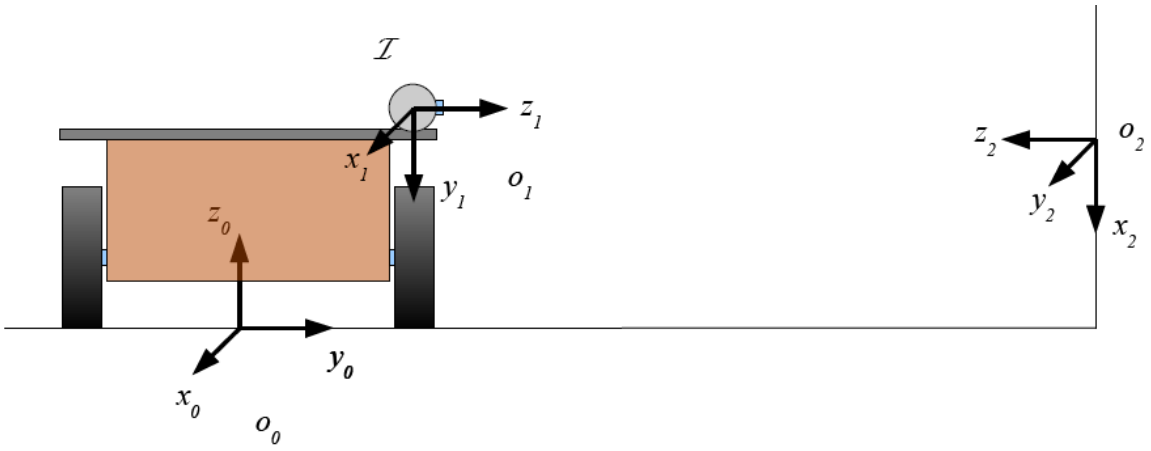


**Figure B.1: Reference frames**

Let $H_b^a \in \mathbb{R}^{4\times4}$ be the transformation that transforms a vector from reference frame $o_a$ to $o_b$. This transformation consists of a rotation matrix $R_b^a \in \mathbb{R}^{3\times3}$ and a translation $T_b^a \in \mathbb{R}^3$. The transformation has the following form:

$$H_b^a = \begin{bmatrix} R_b^a & T_b^a \\ 0 & 1 \end{bmatrix} \tag{B.1}$$

For a vector $v_a \in \mathbb{R}^4$, in the reference frame *a*,

$$v_a = \left[x_a, y_a, z_a, 1\right]^T. \tag{B.2}$$

where $x_a, y_a, z_a \in \mathbb{R}$ are with respect to the reference frame *a*. The following equation transforms $v_a$ into reference frame *b*:

$$v_b = H_b^a v_a . \tag{B.3}$$

The Camera Calibration Toolbox gives a transformation from an object within the image to the camera. This transformation is $H_1^2$. Since the transformation from the robot reference frame to the fixed reference frame, $H_2^0$, is already known, the transformation from the robot reference frame to the camera reference frame is given by

$$H_1^0 = H_1^2 H_2^0 . \tag{B.4}$$

To find the linear velocity of the camera, first transform the linear velocity of the robot into the camera's reference frame. Because the camera is a distance $\left\| T_0^1 \right\|$ from the center of the robot, the cross product of the angular velocity and the vector $T_0^1$ must be transformed into the camera's reference frame and added to the previous result. The linear velocity of the camera is

$$v_c = R_1^0 \left( v_r + \omega_r \times T_0^1 \right), \tag{B.5}$$

where $v_r \in \mathbb{R}^3$ is the linear velocity of the robot and $\omega_r \in \mathbb{R}^3$ is the angular velocity of the robot. To transform the angular velocity from the robot reference frame to the camera frame, the following equation is used:

$$\omega_c = R_1^0 \omega_r \tag{B.6}$$

If $R_1^0$ and $T_0^1$ are known, Eq. (B.5) and Eq. (B.6) allow $v_c$ and $\omega_c$ to be found for any position and orientation of the camera with respect to the robot.

Appendix C

Mobile Robot Sinusoidal Trajectory Derivation

This appendix explains how the translational and rotational velocities were obtained for the mobile robot's sinusoidal trajectory. Taking the time derivative of Eq. (2.33) gives

$$\begin{cases} \dot{x}(t) = k \\ \dot{y}(t) = -ak\omega\sin(k\omega t) \end{cases}. \tag{C.1}$$

The *x* and *y* velocities of the robot are

$$\begin{cases} \dot{x}(t) = v_t\cos\phi \\ \dot{y}(t) = v_t\sin\phi \end{cases}, \tag{C.2}$$

where $v_t$ is the translational velocity of the robot and $\phi$ is the orientation of the robot with respect to the *x*-axis. Setting Eq. (C.1) equal to Eq. (C.2) produces the following 2 equations:

$$v_t = \frac{k}{\cos\phi} \tag{C.3}$$

$$\tan\phi = -a\omega\sin(k\omega t) \tag{C.4}$$

Solving for Eq. (C.4) for $\phi$ gives

$$\phi = \tan^{-1}\left[-a\omega\sin(k\omega t)\right]. \tag{C.5}$$

From the time derivative of Eq. (C.5), the rotational velocity $v_\omega$ of the robot for the sinusoid trajectory is

$$v_{\omega} = \frac{-ak\omega^2 \cos(k\omega t)}{a^2\omega^2 \sin^2(k\omega t)+1}. \tag{C.6}$$

From Eq. (C.3) and Eq. (C.5), the translational velocity $v_t$ of the robot for the sinusoidal trajectory is

$$v_t = k\sqrt{a^2\omega^2 \sin^2(k\omega t)+1}. \tag{C.7}$$

A simulation of these equations was used to produce Figure C.1 showing the path of the robot over a period of 40 seconds. In the figure, $k = 0.05m/s$, $a = 0.05m$, and $\omega = 2\pi\ rad/s$. The initial 2D Euclidean position of the mobile robot is $(0,0)$.

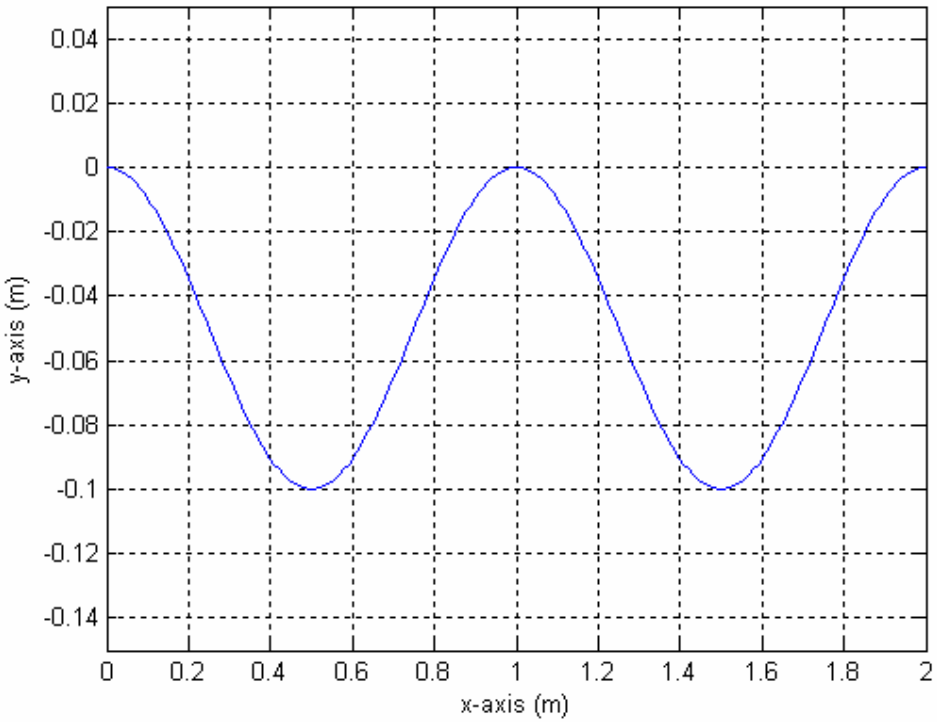


**Figure C.1: Simulated sinusoidal trajectory of the mobile robot**

Appendix D

Proposed Electric Motors and Connectors


**Figure D.1: AC Servomotor – Lexium BPH0552S5UA2C00 Serial #:02764-0008**

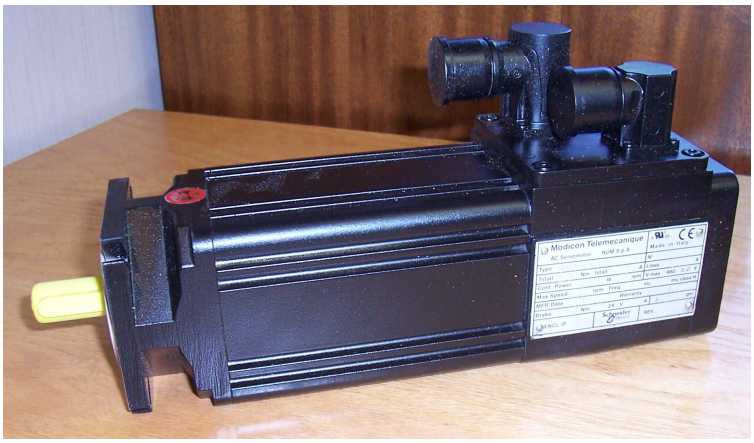


**Figure D.2: AC Servomotor – Lexium BPH0752N5MA2CA1 Serial #:54022704241 (480VAC, three-phase)**

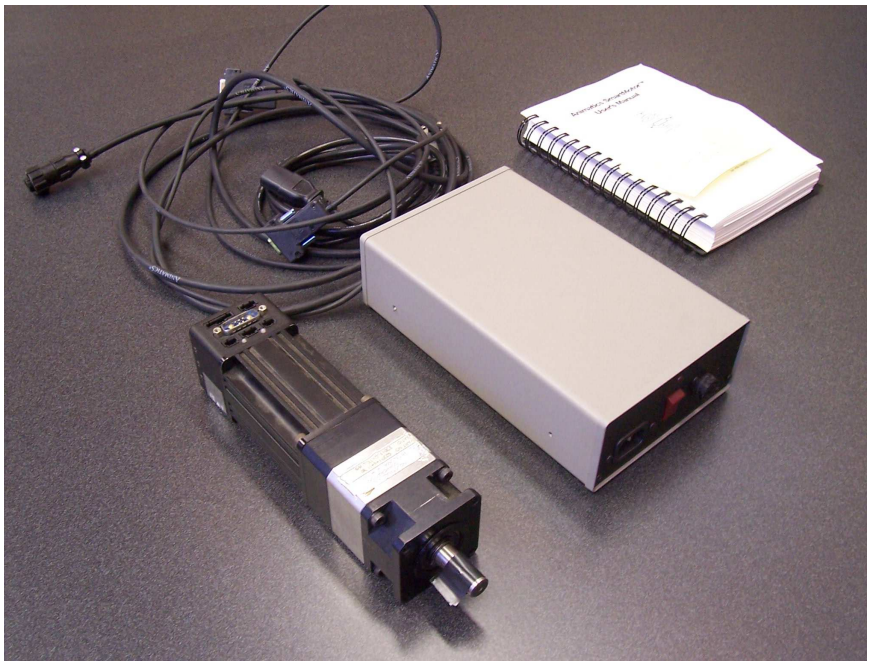

**Figure D.3: DC Servomotor – Animatics SM2340SQ Serial #:E05114 with DC power supply PS42V6A Serial #:P02734**

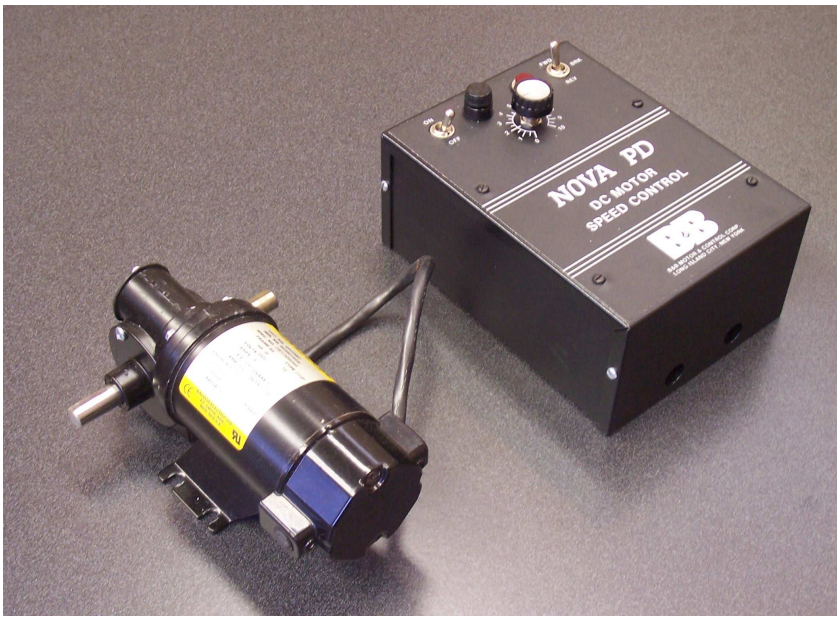


**Figure D.4: DC Servomotor – Baldor GP233007 Serial #:B0308220550 (90V DC, 0.78 A, 0.06 HP) with controller**


**Figure D.5: 1-HP AC induction motor – GE 5KE143BC205 Serial #:2016200219**

**Figure D.6: 1/4-HP AC induction motor – Reliance Electric B79B8940M-KE Serial #:6C2495-92-A**

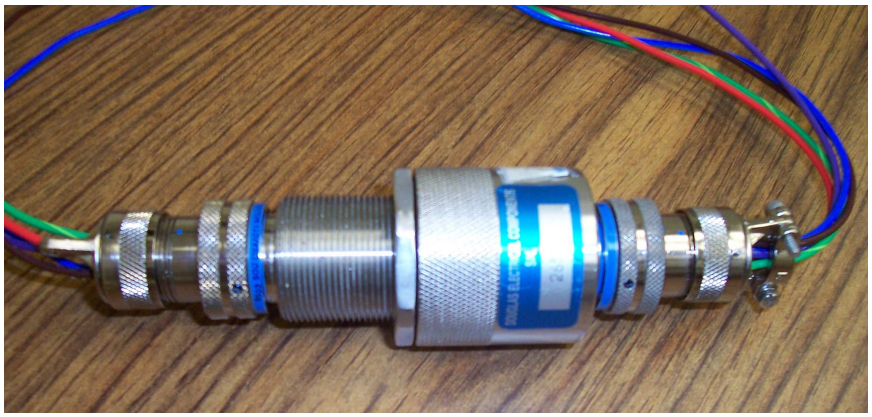

**Figure D.7: Hermetic connector – Douglass POTCON™ 24240 Serial#: 26515**

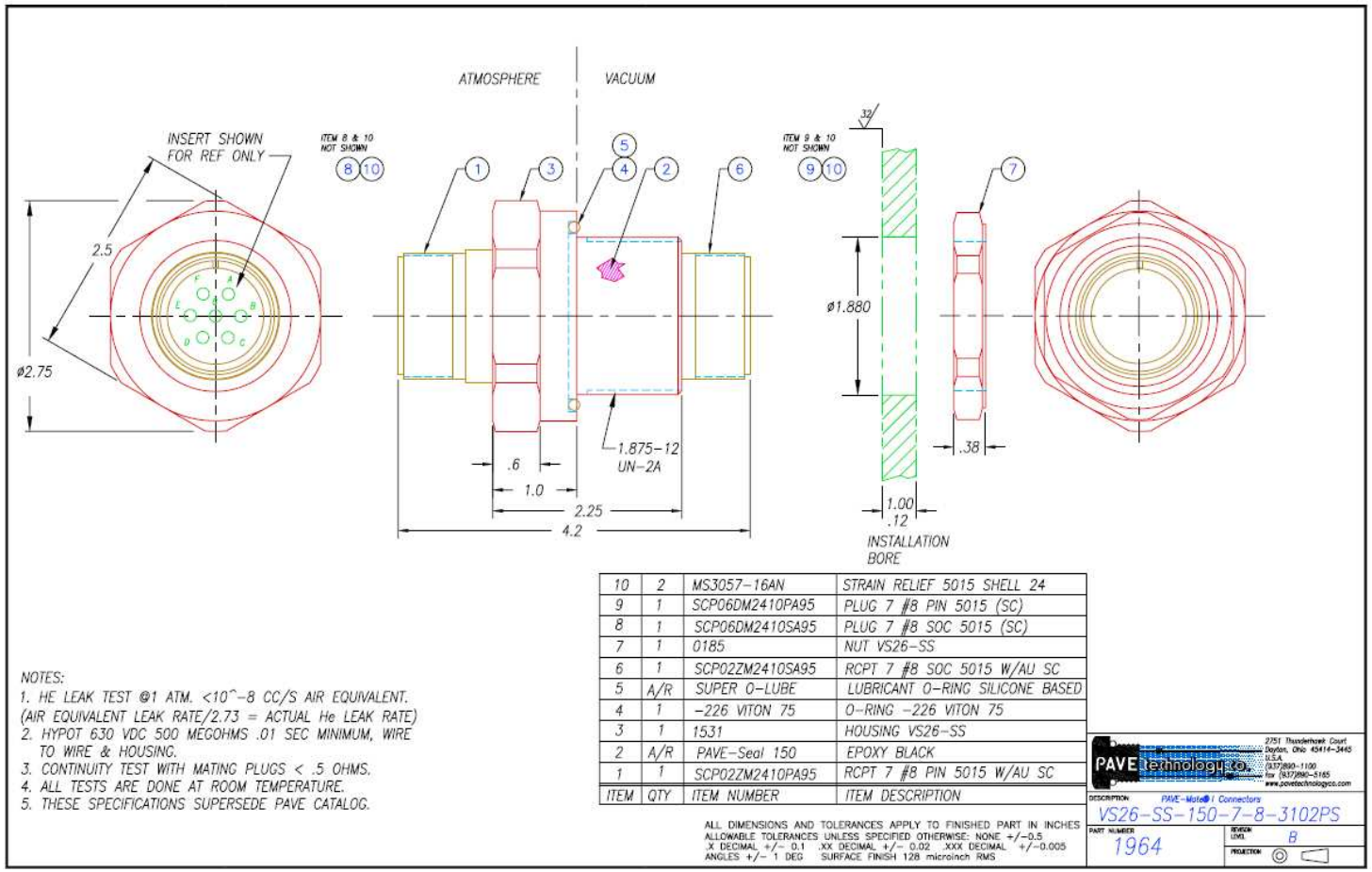


**Figure D.8: Hermetically-sealed bulkhead connector – Pave Technologies #1964**

REFERENCES


[1] J. Oliensis, "A critique of structure from motion algorithms," *Computer Vision and Image Understanding*, vol. 80, no. 2, pp. 172–214, 2000.

[2] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure from controlled motion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 5, pp. 492–504, May 1996.

[3] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 4, pp. 523–535, Apr. 2002.

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," *IEEE Robot. Automat. Mag.*, vol. 13, no. 3, pp. 99–108, 2006.

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, June 2007.

[6] T. Fukao, K. Fujitani, and T. Kanade, "An autonomous blimp for a surveillance system," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, 2003, pp. 1820–1825.

[7] T. Kanade, O. Amidi, and Q. Ke, "Real-time and 3d vision for autonomous small and micro air vehicles," in *Proc. IEEE Int. Conf. Decision and Control*, Dec. 2004, pp. 1655–1662.

[8] J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Visionbased target localization from a fixed-wing miniature air vehicle," in *Proc. American Control Conf.*, Minneapolis, MN, 2006, pp. 2862–2867.

[9] V. N. Dobrokhodov, I. I. Kaminer, K. D. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small uavs," in *Proc. American Control Conf.*, Minneapolis, MN, 2006, pp. 1428–1433.

[10] J.-H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *Proc. IEEE Int. Conf. Robots and Automation*, Taipei, Taiwan, 2003, pp. 406–411.

[11] I.-K. Jung and S. Lacroix, “High resolution terrain mapping using low altitude aerial stereo imagery,” in *Proc. IEEE Int. Conf. Computer Vision*, Nice, France, 2003, pp. 1820–1825.

[12] I. Miyagawa and K. Arakawa, “Motion and shape recovery based on iterative stabilization for modest deviation from planar motion,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 7, pp. 1176–1181, July 2006.

[13] S. J. Julier and J. K. Uhlmann, “A counter example to the theory of simultaneous localization and map building,” in *Proc. IEEE Int. Conf. Robots and Automation*, Seoul, Korea, 2001, pp. 4238–4243.

[14] M. Jankovic and B. K. Ghosh, “Visually guided ranging from observations of points, lines and curves via an identifier based nonlinear observer,” *Systems and Control Letters*, vol. 25, pp. 63–73, 1995.

[15] X. Chen and H. Kano, “State observer for a class of nonlinear systems and its application to machine vision,” *IEEE Trans. Automat. Contr.*, vol. 49, no. 11, pp. 2085–2091, Nov. 2004.

[16] X. Hu and T. Ersson, “Active state estimation of nonlinear systems,” *Automatica*, vol. 40, p. 2075 2082, 2004.

[17] V. K. Chitrakaran, D. M. Dawson, J. Chen, and H. Kannan, “Velocity and structure estimation of a moving object using a monocular camera,” in *Proc. American Control Conf.*, Minneapolis, MN, 2006, pp. 5159–5164.

[18] V. K. Chitrakaran and D. M. Dawson, “A lyapunov-based method for estimation of euclidean position of static features using a single camera,” in *Proc. American Control Conf.*, New York, NY, 2007, to appear.

[19] W. E. Dixon, Y. Fang, D. M. Dawson, and T. J. Flynn, “Range identification for perspective vision systems,” *IEEE Trans. Automat. Contr.*, vol. 48, no. 12, pp. 2232–2238, 2003.

[20] D. Karagiannis and A. Astolfi, “A new solution to the problem of range identification in perspective vision systems,” *IEEE Trans. Automat. Contr.*, vol. 50, no. 12, pp. 2074–2077, 2005.

[21] O. Faugeras, *Three-Dimensional Computer Vision*. Cambridge, Massachusetts: MIT Press, 1993.

[22] V. Chitrakaran, D. M. Dawson, W. E. Dixon, and J. Chen, “Identification of a moving objects velocity with a fixed camera,” *Automatica*, vol. 41, no. 3, pp. 553–562, Mar. 2005.

[23] E. Malis and F. Chaumette, “2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement,” *Int. Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.

[24] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, “Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension,” *IEEE Trans. Contr. Syst. Technol.*, vol. 13, no. 5, pp. 814–825, Sept. 2005.

[25] Open source computer vision library. [Online]. Available: http://www.intel.com/technology/computing/opencv/

[26] Camera Calibration Toolbox for Matlab®. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

[27] D.J. Marne, *National Electrical Safety Code Handbook*, McGraw Hill, New York, 2002.

[28] J.M. Meek and J. D. Craggs, *Electrical Breakdown of Gases*, John Wiley and Sons, New York, 1978.

[29] H. Kirkici. “Electrical Insulation in Space Environment,” in Proc. CEIDP, 2001, p 16-19.

[30] D.G. Kasten, X. Liu, S.A. Sebo, D.F. Grosjean, and D.L. Schweickard, “Partial Discharge Characteristics at Low Pressures in Dry Air and Argon.” Conference Record of IEEE International Symposium on Electrical Insulation, Conference Record of IEEE International Symposium on Electrical Insulation, 2004, p 540-543.

[31] S.J. Swencki, D.B. Burns, J.E. Smith, G.E. Wetzel, D.D. Roybal, and Daleep C. Mohla, “Electrical Safety, Arc Flash Hazards, and ‘The Standards’ A Comprehensive Overview,” IEEE Industry Applications Society 52nd Annual Petroleum and Chemical Industry Conference, 2005, p 343-358.

[32] I.P. Nazarenko and I.G. Panevin, “Simplified Method of Calculating the Electrical Conductivity, Electronic Thermal Conductivity, and Thermodiffusion of Argon,” Teplofizika Vysokikh Temperatur (English translation), v 27, n 3, Nov 1989, p 374-381.

[33] N.B. Varagaftik, *Tables on the Thermophysical Properties of Liquids and Gases*, Hemisphere Pub., Washington D.C., 1975.